

Inteligentni tutorski sustavi za poučavanje programiranja

Karlović, Marko

Undergraduate thesis / Završni rad

2023

Degree Grantor / Ustanova koja je dodijelila akademski / stručni stupanj: **University of Pula / Sveučilište Jurja Dobrile u Puli**

Permanent link / Trajna poveznica: <https://um.nsk.hr/um:nbn:hr:137:601531>

Rights / Prava: [In copyright](#)/[Zaštićeno autorskim pravom.](#)

Download date / Datum preuzimanja: **2024-08-03**



Repository / Repozitorij:

[Digital Repository Juraj Dobrila University of Pula](#)



Sveučilište Jurja Dobrile u Puli

Fakultet Informatike u Puli

MARKO KARLOVIĆ

**INTELIGENTNI TUTORSKI SUSTAVI ZA
POUČAVANJE PROGRAMIRANJA**

Završni rad

U Puli, rujan, 2023.

Sveučilište Jurja Dobrile u Puli

Smjer Informatika

MARKO KARLOVIĆ

**INTELIGENTNI TUTORSKI SUSTAVI ZA
POUČAVANJE PROGRAMIRANJA**

Završni rad

JMBAG: 0303075567, Izvanredni student

Studijski smjer: Informatika

Predmet: Informatizacija uredskog poslovanja

Znanstveno područje: Društvene znanosti

Znanstveno polje: Informacijske i komunikacijske znanosti

Znanstvena grana: Informacijski sustavi i informatologija

Mentor: izv. prof. dr. sc. Snježana Babić

IZJAVA O AKADEMSKOJ ČESTITOSTI

Ja, dolje potpisani Marko Karlović, kandidat za prvostupnika Informatike, ovime izjavljujem da je ovaj Završni rad rezultat isključivo mogega vlastitog rada, da se temelji na mojim istraživanjima te da se oslanja na objavljenu literaturu kao što to pokazuju korištene bilješke i bibliografija. Izjavljujem da niti jedan dio Završnog rada nije napisan na nedozvoljen način, odnosno da je prepisan iz kojega necitiranog rada, te da ikoji dio rada krši bilo čija autorska prava. Izjavljujem, također, da nijedan dio rada nije iskorišten za koji drugi rad pri bilo kojoj drugoj visokoškolskoj, znanstvenoj ili radnoj ustanovi

Student

Marko Karlović

U Puli, 2023. godine

IZJAVA O KORIŠTENJU AUTORSKOG DJELA

Ja, Marko Karlović dajem odobrenje Sveučilištu Jurja Dobrile u Puli, kao nositelju prava iskorištavanja, da moj završni rad pod nazivom Uloga umjetne inteligencije u razvoju weba koristi na način da gore navedeno autorsko djelo, kao cjeloviti tekst trajno objavi u javnoj internetskoj bazi Sveučilišne knjižnice Sveučilišta Jurja Dobrile u Puli te kopira u javnu internetsku bazu završnih radova Nacionalne i sveučilišne knjižnice (stavljanje na raspolaganje javnosti), sve u skladu s Zakonom o autorskom pravu i drugim srodnim pravima i dobrom akademskom praksom, a radi promicanja otvorenoga, slobodnoga pristupa znanstvenim informacijama. Za korištenje autorskog djela na gore navedeni način ne potražujem naknadu.

Student

Marko Karlović

U Puli, 2023. godine

1. UVOD	1
2. INTELIGENTNI TUTORSKI SUSTAVI.....	3
2.1 Opći pojam Inteligentnog tutorskog sustava.....	3
2.2. Struktura Inteligentnog tutorskog sustava	5
2.3 Faze dizajna i razvoja.....	9
3. INTELIGENTNI TUTORSKI SUSTAVI ZA PROGRAMIRANJE	12
3.1 Specifični zahtjevi programiranja	12
3.2 SQL-Tutor.....	13
3.3 Sustav BITS	14
3.3 C-tutor	14
3.4 „A Novel Intelligent tutoring system for learning programing“	15
3.5 CPP TUTOR.....	18
3.6 Inteligentni tutorski sustav za C# programski jezik	24
3.7 Inteligentni tutorski sustav za poučavanje programiranja uz korištenje dinamičkih grafičkih vizualizacija	32
4. PREDNOSTI I OGRANIČENJA INTELIGENTNIH TUTORSKIH SUSTAVA ZA POUČAVANJE PROGRAMIRANJA.....	36
5. BUDUĆI RAZVOJ INTELIGENTNIH TUTORSKIH SUSTAVA ZA POUČAVANJE PROGRAMIRANJA	39
6. ZAKLJUČAK.....	43
SAŽETAK	44
ABSTRACT	45
LITERATURA	46
POPIS SLIKA	53

1. UVOD

Visokoškolske ustanove još uvijek prakticiraju standardni formalni način obrazovanja i ono je trenutno najzastupljenije. Edukatori igraju aktivnu ulogu u ovom obrazovnom modelu, prenoseći svoje vještine učenicima, koji informacije dobivaju pasivno (Akyuz, 2020).

S današnjim brzim tehnološkim napretkom, računalno učenje se kombinira s tehnikama umjetne inteligencije kako bi se stvorili personaliziraniji obrazovni sustavi (Al-Shawwa & Alshawwa, 2020).

Inteligentni tutorski sustavi nisu neka nategnuta futuristička ideja. U praktičnom smislu, oni već postoje. Iako su neobični, sposobni su funkcionirati bez prisutnosti instruktora i mogu pomoći učeniku korištenjem različitih algoritama. Sve veći utjecaj tehnologije mijenja način na koji ljudi uče i kako institucije podučavaju. Ljudi koriste Internet kako bi zatražili online poduku s bilo kojeg mjesta i u bilo koje vrijeme. To je rezultiralo razvojem mnoštva platformi za online učenje koje pružaju cjelovita i učinkovita obrazovna rješenja koja su u potpunosti online. Njihova važnost i doprinos, neupitne su, a očite u scenarijima lockdown-a koje smo nedavno proživljavali zbog pandemije covid-19. Tada je pružanje i promicanje sustava e-učenja bio jedan od glavnih problema svih obrazovnih ustanova. U slučaju nastave programiranja, ona je još teža kada se ne može provoditi uživo, licem u lice, na standardan način učenik-učitelj, jedan-na više ili još bolje ako postoji mogućnost i resursi jedan na jedan jer zahtijeva puno više vremena i predanosti učenika, ali i učitelja ako ga ima. S ovakvim sustavima podiže se efikasnost koja nije pronađena unatoč promicanju e-učenja i istraživanjima u koja su se uključili istraživači iz cijelog svijeta u nedavno proživljenoj zdravstvenoj krizi corona virusa (pandemije covid-19) i čini se da je nemoguće pronaći metodu poučavanja koja bi bila učinkovita za sve učenike. COVID-19 je obrazovnoj instituciji postavio niz jedinstvenih izazova. Suočeni s globalnom pandemijom, pristup tradicionalnim metodama učenja je ograničen. Kao rezultat toga, studenti kasne u učenju. Osobito oni koji pohađaju kolegije programiranja gdje propuštaju satove vježbi. Sada je tek ovaj problem prešao na zumiranje zbog pandemije COVID-19, stoga nije praktično za mentora da gleda pojedinačne slučajeve kako bi radio i da svakom učeniku daje trenutnu povratnu informaciju o tome piše li učenik ispravno svoj kod.

Ove praktične vježbe toliko su važne jer studentima ne daju samo praktično iskustvo s kodiranjem, već i trenutnu povratnu informaciju i smjernice od njihovih mentora.

U ovom radu opisuju se i analiziraju Inteligentni tutorski sustavi kao rješenje problema dostupnosti, efikasnosti, cjeloživotnog-visokog obrazovanja na teorijski način uz pomoć provedenih studija nekih od istraživača s ciljem promidžbe Inteligentnih tutorskih sustava i nekorištenja Inteligentnih tutorskih sustava u visokoškolskim ustanovama jer Inteligentni tutorski sustav nije tu da zamijeni nastavnika nego potpomogne u kvaliteti, efikasnosti i dostupnosti obrazovanja. Cilj je istražiti i opisati faze njihova razvoja i glavne modele od kojih se sačinjava arhitektura, kako se polazi nastava programiranja jednom kad se implementiraju na temelju studija nekih od istraživača, te utvrditi njihove prednosti i nedostatke. Predložit će se budući razvoj s prijedlozima za poboljšanje dosadašnjih.

Iako mali broj provedenih studija, te iako su one donijele velik broj zanimljivih uvida o poboljšanju cjeloživotnog, uvijek dostupnog, efikasnog visokog obrazovanja, mali je broj Inteligentnih tutorskih sustava koji se zapravo koristi (Sharma, Harkishan, 2022).

2. INTELIGENTNI TUTORSKI SUSTAVI

Termini tutor i mentor vežu se za neku osobu koja daje stručnu pomoć u nekom određenom području, u svladavanju neke određene vještine, prenošenju znanja iz cijelog tog područja u procesu učenja. Ako se to prenese na kontekst Umjetne inteligencije i Informacijsko-komunikacijske tehnologije, inteligentni tutorski sustav (Intelligent Tutoring Systems, ITS) je softver koji simulira ponašanje tutora/mentora/učitelja/nastavnika ovisno o njegovoj implementaciji. Inteligentni tutorski sustav je softver koji pruža učeniku individualnu, uvijek dostupnu podršku i pomoć pri učenju i dodatnu pomoć uz postavljanje pitanja učeniku, rasčlanjivanje njihovih odgovora ili zadataka i na temelju toga nudi povratne informacije ili instrukcije koje su za učenje i pomoć u razumijevanju sadržaja. Ono po čemu se Inteligentni tutorski sustav razlikuje od ostalih, jest da za svakog učenika protumači njegov odgovor i u skladu s njim reagira, dakle ne prepoznaje samo da li je učenik odgovorio točno/netočno već u kojem koraku je pogriješio, te na temelju toga šalje povratne informacije i instrukcije koje su prilagođene svakom učeniku zasebno. Također, u isto vrijeme može raditi sa velikim brojem učenika odjednom (Kuleto, 2017).

2.1 Opći pojam Inteligentnog tutorskog sustava

Inteligentni tutorski sustavi usmjereni su na stručnost učenika koja proizlazi iz interpretacije njihovog kontakta s predmetom. Na temelju jedne od analiziranih studija, agent s umjetnom inteligencijom određuje kognitivne profile učenika i prilagođava ih potrebama učenja koje odgovaraju tim profilima (Binh, Trung, 2021). Na primjer, pedagoški materijal dostupan studentima prilagođen je njihovim potrebama učenja kako bi se optimizirao. Da bi se zaključila učenikova kognitivna stanja, kao što je njegov ili njezin stupanj kompetentnosti ili stručnosti, pronicljivi uređaj za podučavanje mora biti sposoban dijagnosticirati ponašanje učenika. Ovom pristupu dobro služi korištenje studentskog modela, koji prati kognitivna stanja učenika i kontinuirano ga mijenja (inteligencija sustava) (Kumar & Ahuja, 2020).

Važno je spomenuti da je Inteligentni tutorski sustav u istraživanjima do sad postigao veliku efikasnost i u postavljanju pretpostavki kako učenik uči, što u zadatku ne razumije. Njegove instrukcije, savjeti, efikasnije su kasnije nego ukazivanje na to što

učenik ne zna. Prilagodбом brzine prema učeniku, svakome zasebno i njegovu načinu rada, bez vremenskih ograničenja, efikasnost učenja je velika. Također ovdje kod efikasnosti učenja važno je i spomenuti da kompjuterski tutor ne izaziva anksioznost koju neki ljudi mogu osjećati, kod učenika, kod priznavanja pogreške, i s time je lakše učeniku razumjeti neki koncept nego u učionici, u socijalnoj situaciji u učionici ili uz pritisak realnog starijeg nastavnika (Kuleto, 2017).

Prema Dadiću, (2023) oblikovanje inteligentnih tutorskih sustava za poučavanje programiranja temelji se na: umjetnoj inteligenciji, teoriji inteligentnih tutorskih sustava, kognitivnoj psihologiji učenja programiranja i prevođenju programskih jezika. Specifična značajka tutorskih sustava za poučavanje programiranja je lociranje i objašnjenje semantičkih pogrešaka ovisnih o zadatku, predstavlja razlike i relacije između napisanih i programa koje je učenik namjeravao napisati.

S aspekta umjetne inteligencije - inteligentni tutorski sustav promatra se kao inteligentan agent. Sustav reagira na učenikovo znanje tako da mu prikaže nastavne jedinice čiji sadržaj nedostaje u njegovu znanju. Korisnost djelovanja mjeri se postignućem učenika u rješavanju programskih problema. Djelovanje sustava očituje se kroz dvije razine povratnih veza: strateška i operativna.

- Strateška – pronalazi razliku između učenikovog znanja u odnosu na ciljno
- Operativna – pronalazi razliku između postignutog i očekivanog rješenja u tekućem zadatku

.S aspekta kognitivne psihologije učenja - razumijevanje svojstava programskog jezika znatno je naprednije ako se izvođenje programskih funkcija opiše prirodnim jezikom.

Pristupi tutorskog poučavanja:

- Preliminarno – tutor unaprijed pripremi učenika za rješavanje zadatka, konačno rješenje vrednuje se kao točno ili netočno, bez objašnjenja pogrešaka.
- Interaktivno – Učenik sa svojim dosadašnjim znanjem područja pristupa rješavanju zadatka bez uputa a tutor intervenira ako učenik odustaje ili čim počne nepovratno odstupati od putanje do točnog rješenja.

- Refleksivno – Učenik samostalno pristupa rješavanju zadatka, a tutor bez intervencija, nakon predaje rješenja, pregledava rješenje i objašnjava počinjene greške.

Ako usporedimo pristupe tutorskog poučavanja može se uočiti da u preliminarnom poučavanju nema operativne povratne veze a u refleksivnom može biti zakašnjela i uzrokovati protezanje pogreške tijekom programa i potencijalno stvaranje novih (Dadić, 2023).

2.2. Struktura Inteligentnog tutorskog sustava

Inteligentni tutorski sustav funkcionira tako da svaki od Inteligentnih tutorskih sustava najvjerojatnije nema istu strukturu i ne funkcionira niti slično, no u stručnoj literaturi postoji konsenzus o 4 najčešća glavna elementa strukture Inteligentnog tutorskog sustava. To su: modul područnog eksperta, model učenika, modul studenta i komunikacijski modul (Nwana, 1990).

Model područnog eksperta ili domene

Model područnog eksperta ili domene može biti izgrađen na teoriji učenja, kao što je teorija ACT-R koja pokušava uzeti u obzir sve moguće korake potrebne za rješavanje problema. Točnije, ovaj model sadrži koncepte, pravila i strategije rješavanja problema domene koja se uči. Može ispuniti nekoliko uloga: kao izvor stručnog znanja, standard za ocjenjivanje učenikove izvedbe ili za otkrivanje pogrešaka itd. . (Nkambou i sur., 2010). Drugi pristup razvoju modela domene temelji se na Teoriji učenja iz pogrešaka u izvedbi Stellana Ohlssona (Ohlsson, 1996). poznatom kao modeliranje temeljeno na ograničenjima (CBM) (Ohlsson, 1992). Ovaj model “zna” korektno riješiti specificirani problem iz područja poučavanja. Osim toga, raspoložbe zbirkom karakterističnih pogrešaka, u stanju je dijagnosticirati učenikovo rješenje i u tom slučaju, model domene može se gledati kao skup ograničenja za točna rješenja (Mitrovic, Ohlsson, 2006).

Model studenta

Model studenta predstavlja trenutno znanje učenika kao bazu podataka (Dadic, 2008).

Studentski model može se smatrati preklapanjem na modelu domene. Smatra se ključnom komponentom ITS-a koja posebnu pozornost posvećuje kognitivnim stanjima učenika i njihovoj evoluciji kako proces učenja napreduje.

Kad god studentski model odstupa od modela domene, sustav prepoznaje ili označava da je došlo do pogreške (Mitrovic, Ohlsson, 1999). Tutori koji se temelje na ograničenjima daju negativne povratne informacije (tj. povratne informacije o pogreškama) i također pozitivne povratne informacije (Mitrovic, Ohlsson, Barrow, 2013).

Karakteristika većine inteligentnih tutorskih sustava je da zaključuju o tome kako učenik razumije dano gradivo i to iskoriste da bi poučavanje prilagodili učenikovim potrebama. Ovakvo zaključivanje još se zove i postavljanje dijagnoze. On mora zaključiti što je učenik mislio i čime se bavio tijekom učenja. Trenutačno stanje učenikova znanja predstavljeno je modelom učenika. Učenikov model je skup podataka, a dijagnoza je proces manipulacije tim podacima (Stankov, 2001).

Model računalnog mentora

Model računalnog mentora na temelju modela studenta odlučuje o napredovanju nastavnog sadržaja, bira primjerene zadatke, objašnjava pogreške. Mentor bi morao znati što učenik zna, koja je razina njegovih sposobnosti, njegovo predznanje. Mentor mora odabrati hoće li zanemarivati pogrešku koju je učenik napravio, istaknuti tu pogrešku, ispraviti pogrešku ili na neki način voditi učenika prema prepoznavanju i ispravljanju pogrešaka vlastitim snagama (Dadic, 2008).

Prihvća informacije iz modela domene i studenta te donosi odluke o strategijama podučavanja. U bilo kojem trenutku u procesu rješavanja problema učenik može zatražiti smjernice o tome što učiniti sljedeće, u odnosu na svoju trenutnu lokaciju u modelu. Osim toga, sustav prepoznaje kada je učenik odstupio od pravila modela i pruža pravovremenu povratnu informaciju za učenika, što rezultira kraćim vremenskim

razdobljem za postizanje ciljanih vještina (Anderson, Koedinger, 1997). Model mentora može sadržavati nekoliko stotina proizvodnih pravila za koja se može reći da postoje u jednom od dva stanja, naučenog ili nenaučenog. Svaki put kada učenik uspješno primijeni pravilo na problem, sustav ažurira procjenu vjerojatnosti da je učenik naučio pravilo. Sustav nastavlja vježbati učenike na vježbama koje zahtijevaju učinkovitu primjenu pravila sve dok vjerojatnost da je pravilo naučeno ne dosegne najmanje 95% vjerojatnosti (Corbett et al., 2008).

Modul komunikacije

Modul komunikacije jest korisničko sučelje i podržava razmjenu informacija učenika sa sustavom.

Jasno je nakon mnogo provedenih istraživanja da se zahtijeva izgradnja ovog modula kao zasebnog i da je ovaj modul jedan od glavnih jer omogućuje komunikaciju prirodnim jezikom, što se već i ranije u radu navelo po radu Dadića iz 2023. da učenik znatno naprednije razumije svojstva programskog jezika ako se izvođenje programskih funkcija opiše prirodnim jezikom.

Ako je sučelje loše oblikovano, efikasnost obrazovnih sesija bit će lošija. Suprotno tome, dobro oblikovano sučelje učenika poboljšava sposobnosti ITS-a. Ono prema Stankovu iz knjige Inteligentni tutorski sustavi 3. ili C dio iz 2014. navodi učenika da bude aktivni sudionik obrazovnog procesa, a ne samo pasivni promatrač. Predstavljanje znanja mora aktivirati nekoliko učenikovih osjetila (Stankov, 2014).

Nwana (1990) izjavljuje: "Gotovo je rijetkost pronaći dva inteligentna tutorska sustava temeljena na istoj arhitekturi što proizlazi iz eksperimentalne prirode rada u tom području".

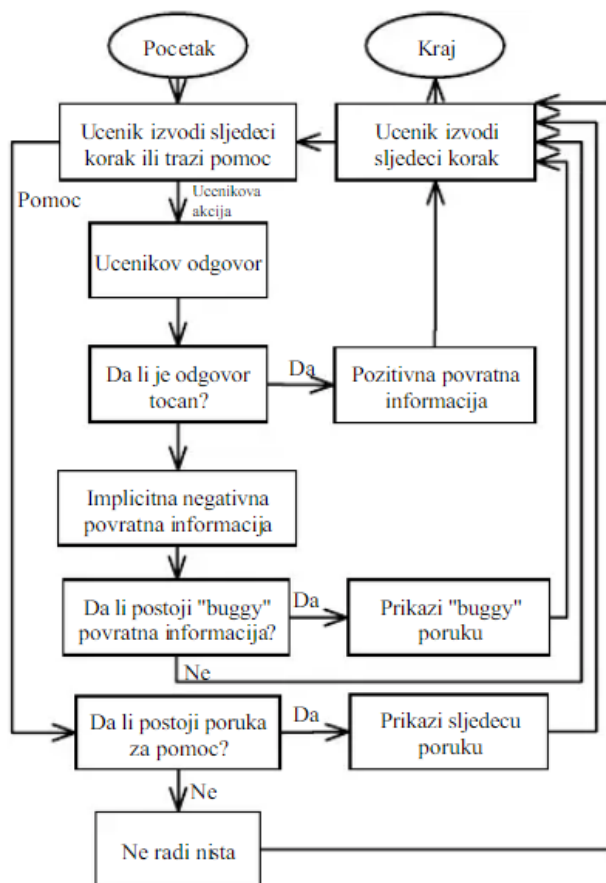
Praćenje znanja

Praćenje znanja ili model traganja prati napredak učenika i izrađuje profil uspješnosti učenika u odnosu na skup pravila u domeni.

Slavomir Stankov i dr., 2001. opisuju model traganja u autorskoj ljusci na temelju koje su provodili i gradili istraživanja na ovaj način:

„Algoritam za dijagnostiku aktivira sva predložena pravila, te postiže skup mogućih sljedećih stanja. Jedno od ovih stanja mora odgovarati stanju koje je učenik proizveo. Ime “model traganja” proizlazi iz činjenice da dijagnostički program prati izvršavanje u modelu i to uspoređuje s aktivnosti učenika“ (Stankov i dr., 2001).

Prema Stankovu i drugima iz 2001. godine slika 1 prikazuje njihov algoritam u modelu traganja. Radio je na način da je primao neke ulazne podatke (1) znanje iz domene, (2) problem zadatka koji učenik rješava, (3) učenikov odgovor. Izlazni podaci bili su (1) točno ili netočno ovisno o tome da li je algoritam pronašao učenikov odgovor, (2) ako je odgovor bio pronađen, izlazni podaci bili su niz izvedbi koje su mogle biti učenikov odgovor, ako odgovor nije bio pronađen, znači da ga algoritam nije mogao pronaći nijednu izvedbu koja bi mogla biti učenikov odgovor i učenikov odgovor se smatrao neinterpretiranim. No, postoje i izvedbe s kojima se pronalaze učenikovi pogrešni odgovori zvane „buggy“ izvedbe, sustavu su razumljive a koriste se da bi učenikov odgovor imao sustavu smisla, čak i kada bi učenik napravio nekoliko grešaka tijekom izvedbe. Sustav je učenika obavijestio o njegovim greškama i generirao povratnu informaciju na dva načina: (1) „buggy“ povratna informacija i (2) savjet ili pomoć. “buggy” pravilo generiralo je poruku o napravljenoj pogrešci, a savjet ili pomoć je učenik mogao zatražiti i dobiti ili bi se dobili kad bi sustav procijenio da su potrebni. Model traganja i algoritam bi se pokrenuli nakon svake učenikove akcije a služi za nadgledanje učenika kad rješava problem.



Slika 1 Prikaz tradicionalne arhitekture modela traganja. (Izvor: Stankov i dr., 2001).

2.3 Faze dizajna i razvoja

Inteligentni tutorski sustavi su implementirani na razne načine, korištenjem alata kao što su LISP, CGI i drugi (González et al., 2013).

Osim neusklađenosti među arhitekturama inteligentnih tutorskih sustava, razvoj inteligentnih tutorskih sustava gotovo je isti kao bilo koji proces dizajna nastave. Corbett i sur. (1997) sažeo je dizajn i razvoj inteligentnih tutorskih sustava koji se sastoji od četiri iterativne faze: (1) procjena potreba, (2) analiza kognitivnih zadataka, (3) početna implementacija mentora i (4) evaluacija.

Prva faza, poznata je kao procjena potreba. Cilj je specificirati ciljeve učenja i ocrtati opći plan za nastavni plan i program. Pritom se treba pozabaviti (1) vjerojatnošću da je učenik sposoban riješiti probleme, (2) vrijeme koje je potrebno da se postigne ova

razina uspješnosti i (3) vjerojatnost da će učenik aktivno koristiti ovo znanje u budućnosti. Drugi važan aspekt koji zahtijeva analizu je isplativost sučelja.

Druga faza, kognitivna analiza zadataka, detaljan je pristup programiranju ekspertnih sustava s ciljem razvoja valjanog računalnog modela. Glavne metode za razvoj modela domene uključuju: (1) intervjuiranje stručnjaka iz domene, (2) provođenje studija protokola "razmišljaj naglas" sa stručnjacima iz domene, (3) provođenje studija "razmišljaj naglas" s početnicima i (4) promatranje nastave i učenja ponašanje.

Treća faza, početna implementacija mentora, uključuje postavljanje okruženja za rješavanje problema kako bi se omogućio i podržao autentičan proces učenja. Nakon ove faze slijedi niz aktivnosti evaluacije kao posljednja faza koja je opet slična svakom projektu razvoja softvera.

Četvrta faza, evaluacija, uključuje: (1) pilot studije za potvrdu osnovne upotrebljivosti i obrazovnog učinka, (2) formativne evaluacije sustava u razvoju, (3) parametarske studije koje ispituju učinkovitost značajki sustava i (4) evaluacija konačnog učinka mentora, stopa učenja i razine postignuća (Corbett , Koedinger, Anderson, 1997).

Anderson i sur. (1987) opisali su osam načela za dizajn inteligentnog tutora:

1. Predstaviti studentsku kompetenciju kao proizvodni skup.
2. Komunicirajte strukturu cilja koja je u osnovi rješavanja problema.
3. Dajte upute u kontekstu rješavanja problema.
4. Promicati apstraktno razumijevanje znanja o rješavanju problema.
5. Minimizirajte opterećenje radne memorije.
6. Pružite trenutnu povratnu informaciju o pogreškama.
7. Prilagodite veličinu zrna uputa s učenjem.
8. Olakšati uzastopne aproksimacije ciljne vještine.

A kasnije su Corbet i dr. (1997) razradili ta načela ističući sveobuhvatno načelo za koje su vjerovali da upravlja inteligentnim dizajnom nastavnika, a to su načelo nazvali kao: **Načelo 0**: Inteligentni tutorski sustav trebao bi omogućiti učeniku da radi do uspješnog završetka rješavanja problema (Corbett i dr., 1997).

3. INTELIGENTNI TUTORSKI SUSTAVI ZA PROGRAMIRANJE

Programiranje predstavlja višestruke poteškoće za učenike koji ga tek počinju učiti. Npr.: korištenje varijabli, razumijevanje kontrolnih struktura, ispravljanje sintaktičkih pogrešaka, itd... . Konkretno, uključuje stupanj apstrakcije kod učenika koji oni još uvijek nemaju jer ne razumiju pravi učinak koji bi mogla imati promjena izvornog koda programa na njegovo izvođenje ili zato što ne pružaju odgovarajuće rješenja problema korištenjem programskog jezika. Brojne su prednosti korištenja inteligentnih tutorskih sustava ili pomoćnika kao pomoćnih alata u kontekstu učenja programiranja, budući da su usmjereni prema razvoju učinkovitijih personaliziranih procesa poučavanja i učenja . Iz perspektive uloge nastavnika, razvoj ove vrste instrumenata za njih uključuje snažne mogućnosti, među kojima je smanjenje vremena za osmišljavanje sadržaja i promicanje odgovarajućeg znanja. S druge strane, sa stajališta učenika, učenje se odvija na autonoman način (Schez-Sobrinio i dr., 2020).

3.1 Specifični zahtjevi programiranja

Pri razvoju inteligentnog tutorskog sustava za programiranje u nekom jeziku, potrebno je napraviti istraživanje o potrebnim temeljnim vještinama i razvoj „mekanih“ (soft skills) vještina odnosno individualnih nemjerljivih vještina i osobina prijeko-potrebni za napredak, koje će se trebati integrirati u inteligentni tutorski sustav i s kojima će on „znati“ baratati kako bi znao prenijeti to znanje i razvijati ga učeniku za programiranje u istom.

Prema podacima sa sveučilišta u Denveru temeljne i meke vještine su:

Temeljne vještine:

- Znanje o vrstama i strukturama podataka i algoritmima
- Znanje o bazama podataka koje su kao gorivo na koje to što se programira kasnije radi i SQL-u (jeziku koji se koristi za rad s bazama podataka)
- Što je kompajler
- Kako funkcionira Internet
- Naredbeni redak
- Uvjeti i petlje (if, if-else, for, while, ...)
- Objektno – orijentirani način programiranja (objekti, klase)

- Korištenje: Integriranih razvojnih okruženja (eng. short: IDE)
- Računalstva u oblaku
- WEB development
- Spremnici
- Urednici teksta (koda)
- Znanje o korištenju GIT-a

„Soft skills“ :

- Komunikacija
- Timski rad
- Rješavanje problema
- Strpljenje
- Prilagodljivost
- Empatija
- Znatiželja
- Odgovornost
- Upravljanje vremenom

(University of Denver Bootcamps, pristupljeno[20.09.2023]).

Učenje programiranja smatra se problematičnim. Jedan razlog zašto studentima ne ide dobro u programiranju je tradicionalni način učenja unutar predavaonica jer dodaje više stresa studentima u razumijevanju. Za nekoliko učenika, ovaj model podučavanja možda neće privući njihov interes. Kao rezultat toga, neće dati sve od sebe da shvate dani materijal. Međutim, ako se informacije primjenjuju na stvarne probleme, to će povećati interes učenika za učenje i to može povećati njihov trud da se poučava (Al-Bastami, Abu-Naser, 2017).

3.2 SQL-Tutor

SQL-Tutor prvi je ikad ITS temeljen na ograničenjima koji je razvila Intelligent Computer Tutoring Group (ICTG) na Sveučilištu Canterbury , Novi Zeland. SQL-Tutor uči studente kako dohvatiti podatke iz baza podataka pomoću naredbe SQL SELECT (Mitrović, 2003).

3.3 Sustav BITS

Sustav BITS razvijen je na Sveučilištu Regina, Kanada, objavljen je 2006. godine (Butz et al., 2006) . BITS je web based sustav za adaptibilno vođenje učenika nastavnim sadržajem. Na temelju modela učenika može preporučiti ciljeve učenja i generirati najbolji slijed napredovanja. Nalazeći nemogućim egzaktno odrediti učenikovo znanje, autori primjenjuju Bayesian mrežu, kao dobro definiran okvir vjerojatnosnog zaključivanja. BITS nastavni sadržaj posvećen reduciranom jeziku C++ prikazan Bayesian mrežom, te ju prema učenikovim odgovorima na kviz pitanja, ažurira vjerojatnim faktorima znanja (Dadić, 2008).

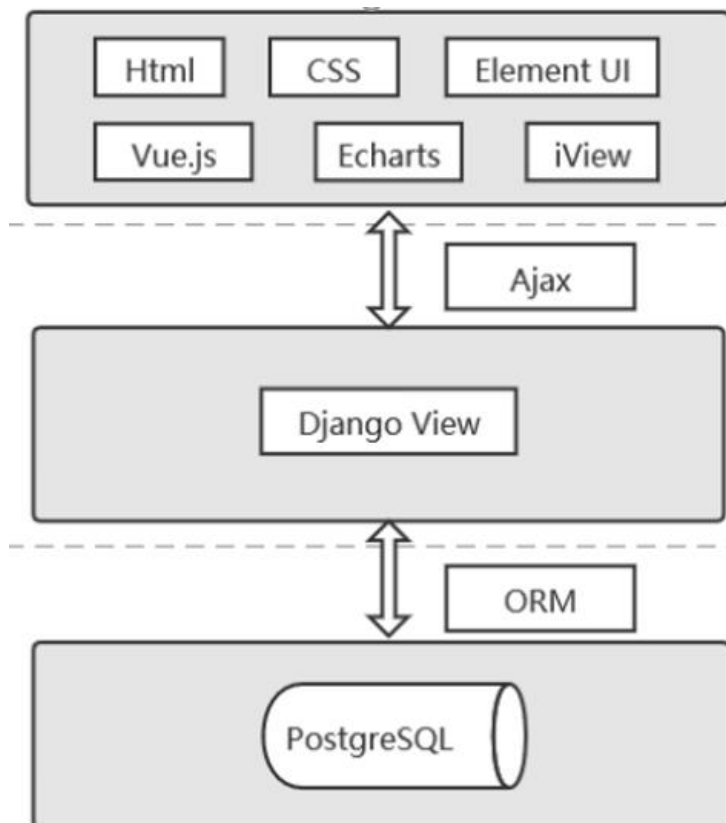
3.3 C-tutor

C-tutor je inteligentni sustav podučavanja za programiranje u C jeziku za početnike. Koristi programski analizator kao najvažniji dio inteligentnog tutorskog sustava za programiranje. U primjeru inteligentnog tutorskog sustava programski analizator je spoj obrnutog inženjeringa i didaktičkog sustava. Budući da program početnika obično sadrži mnogo grešaka, informacija o namjerama programera neizbježna je za prepoznavanje programa s greškama. U C-tutoru namjera programera se izdvaja kao opis problema iz uzorka programa sustavom obrnutog inženjeringa koji se zove GOES (GOal Extraction System). S tim opisom problema učenikove programe prepoznaje didaktički sustav ExBug (Execution-guided deBugger). Kao okruženje za učenje Curriculum Network gradi bazu znanja kao genetske grafikone za podučavanje programiranja. C-Tutor je kompletan ITS koji pruža i analizator programa i okruženje za učenje. Testiran s programima stvarnih učenika, analizator programa daje prihvatljive rezultate prepoznavanja. Analizator programa i okruženje za učenje blisko su povezani tako da studenti mogu učiti C jezik tijekom programiranja (Song et al., 1997).

3.4 „A Novel Intelligent tutoring system for learning programming“

Tako je naslov inteligentnog tutorskog sustava od Meng Wang, Wenjun Wu, Yu Liang iz Kine sa Sveučilišta u Pekingu koji su predstavili na 15. Internacionalnoj konferenciji razvoja i primjene sustava u Rumunjskoj 2020.

Njihova arhitektura inteligentnog tutorskog sustava izgleda ovako:



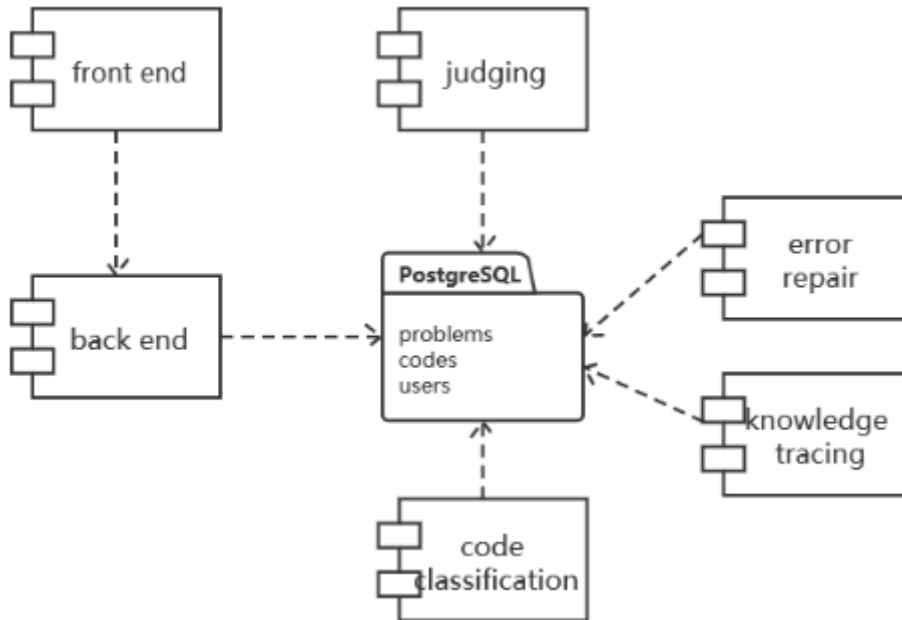
Slika 2. Arhitektura „Novel Intelligent tutoring system for learning programming“. (Izvor: Wang, Wu, Liang, 2020).

Front-end je implementiran u Vue.js framework-u s html-om, css-om i nekim UI-knjižnicama.

Back-end je implementiran u Django framework-u s pythonom i komuniciraju preko AJAX-a. Baza podataka koju koristi je PostgreSQL s kojom komunicira, i barata podacima preko ORM-a (Object relation mapping-a)

Teorija i metoda rada ovog inteligentnog tutorskog sustava bazira se na pretpostavki da su za određeni problem prihvaćena rješenja pohranjena u bazi podataka sustava. Dinamičkom analizom programa prihvaćena rješenja se automatski dijele u nekoliko klasa. U svakoj klasi odabere se program kao specifikacija klase, koji odgovara

rješenju u klasi. Ako se pojavi greška, program neće biti prihvaćen i pokrenut će se algoritam za popravak grešaka, sustav će odabrati naj odgovarajući popravak greške i poslati korisniku odgovarajuću povratnu informaciju.



Slika 3. Komponente „Novel Intelligent tutoring system-a“. (Izvor: Wang, Wu, Liang, 2020).

- Front-end modul predstavlja modul korisničkog sučelja sustava i prikazuje zaslone (stranice) npr. za predaju rješenja.
- Back-end modul prima npr. predano rješenje korisnika, šalje potrebne upite na bazu, analizira i procesuirat podatke i vrati povratnu informaciju na front-end.
- PostgreSQL – baza podataka sprema sve podatke (podatke o korisniku, rješenja zadataka, ...) predstavlja model domene i model studenta.
- Judging modul prosuđuje, tj. proučava predana rješenja korisnika i spremi ih takve u bazu podataka (model studenta-podaci o studentu).
- Code classification modul sadrži točna rješenja problema, klasificira ih i sprema u bazu podataka kao točna rješenja.

CLARA engine - Primjer engina za klasifikaciju koda koji se ujedno koristi i u ovom primjeru inteligentnog tutorskog sustava za podučavanje programiranja. Algoritam za klasifikaciju koda koji koristi CLARA je baziran na podudaranju kod usporedbe koda, ako se predano i točno rješenje iz baze podudaraju znači da pripadaju istoj klasi.

- **Error repair modul** debugira predana kriva rješenja, i kao povratnu informaciju daje savjete. Predstavlja tutorski model.

```

1 def average(a, b, c):
2
3     s = (a+b+c)/2
4     d = s*(s+a)*(s+b)*(s+c)
5     x = d ** 0.78
6
7     print x+1

```

Hint Repair

* Change the assigned value(s) to the variable 'd' at line 4
 * Change the assigned value(s) to the variable 'x' at line 5
 * Change the 'printf' (output) statement at line 7: use template "VARIABLE"

** Change 'print (x + 1)' to 'print x' at 7 (cost=2.0)
 ** Change 'd = (((s * (s + a)) * (s + b)) * (s + c))' to 'd = (((s * (s - a)) * (s - b)) * (s - c))' at 4 (cost=3.0)
 ** Change 'x = (d ** 0.78)' to 'x = (d ** 0.5)' at 5 (cost=1.0)

Slika 4. Primjer davanja savjeta i popravka greške u „Novel Intelligent tutoring system“. (Izvor: Wang, Wu, Liang, 2020).

Slika 4 prikazuje primjer predaje koda s greškom i 2 opcije povratne informacije ITS-a:

- 1) Hint (savjet) koji se može iskoristiti za pregled programa i otkrivanje mjesta greške i savjet za modifikaciju programa i u kojoj liniji kako bi on bio točan ali uz samostalno otkrivanje potrebne modifikacije označene linije kako bi program bio točan.
- 2) Repair (popravi) koja se može koristiti za ispis primjera točno napisanih linija koda.

- **Modul praćenja znanja** dobiva i sadrži podatke iz baze podataka u ovom primjeru predano rješenje korisnika, ali i povijest korisnikovih koraka i povijest već prije predanih rješenja, i vraća vjerojatnost i evaluaciju korisnikova savladavanja lekcije nakon analiziranja i procesuiranja podataka iz korisnikovih

koraka do dolaženja do predanog točnog ili krivog rješenja i komuniciranja sa sustavom. (Wang, Wu, Liang, 2020).

3.5 CPP TUTOR

Arhitektura CPP tutora prema Abu-Naser (2008):

Model domene, baze znanja CPP tutora

U njihovom trenutnom pilot projektu koriste podskup gramatike C++ programskog jezika za podučavanje. Usredotočili su se na sljedeće teme standardnog jezika C++: deklaracija varijabli, operatori, petlje (for petlja i while petlja) i strukture.

Za svaki problem pohranjen u bazi podataka pohranili su jedno moguće rješenje, nekoliko mogućih pogrešaka za određene kategorije i nekoliko mogućih savjeta za svaku pogrešku.

Postoji mnogo mogućnosti za odgovore učenika i sustav ne može jednostavno ispisati sve netočne odgovore učenika. Na primjer, učenik može napisati:

```
c=pow(j,3); ili c=pow(j,2)*j
```

Oba odgovora su u potpunosti točna i sustav treba prepoznati ove vrste odgovora i ne tretirati ih kao netočne odgovore. CPP-Tutor se fokusira na metodologiju kojom student pokušava riješiti problem. Baš kao što je prikazano u ovom primjeru, CPP-Tutor fokusira učenika na problem navodeći mjesto gdje se kod može drugačije zapisati. Na ovaj način se može odvijati učinkovito podučavanje.

Problem: Write a program in C++ which find and print the cubic value for each integer between -5 and +5.

Program specifications: This program can be written using for or while loop.

The solution which is authored by Instructor:

```
#include<iostream.h>
void main() {
    int c, j= -5;
    while (j <= 5) {
        c = j * j * j;
        j++;
        cout << c << endl;
    }
}
```

The student first possible error : (missing #include<iostream.h> in line 1)

```
void main (){
    int c, j= -5;
    while (j <= 5) {
        c = j * j * j;
        j++;
        cout << c << endl;
    }
}
```

The student second possible error: (void is missing in line 2)

```
#include<iostream.h>
main () {
    int c, j= -5;
    while (j <= 5) {
        c = j * j * j;
        j++;
        cout << c << endl;
    }
}
```

The student third possible error: (unknown Int in line 3)

```
#include<iostream.h>
void main() {
    int c, j= -5;
    while (j <= 5) {
        c = j * j * j;
        j++;
        cout << c << endl;
    }
}
```

The student fourth possible error: (undeclared variable c in line 5)

```
#include<iostream.h>
void main() {
    int j= -5;
    while (j <= 5) {
        c = j * j * j;
        j++;
        cout << c << endl;
    }
}
```

The student fifth possible error: (the body of the loop is never executed)

```
#include<iostream.h>
void main() {
    int c, j= -5;
    while (j >= 5) {
        c = j * j * j;
        j++;
        cout << c << endl;
    }
}
```

Slika 5 Primjer zadatka s rješenjem i nekoliko mogućih netočnih rješenja iz CPP tutora. (Izvor: Abu-Naser, 2008).

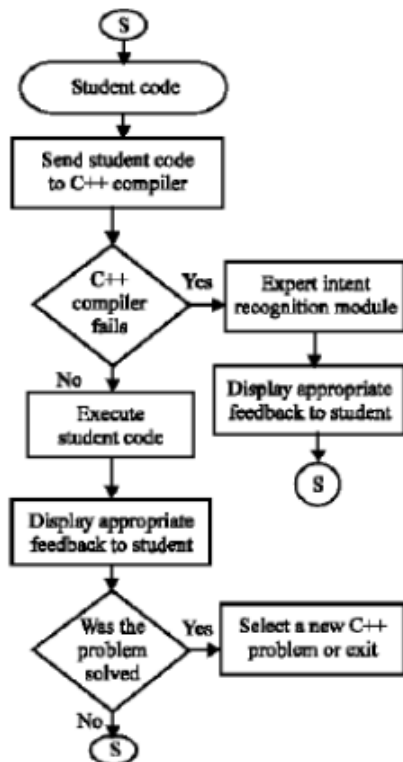
Slika 5 prikazuje pet mogućih krivih sintaksnih greški učenika za zadatak u kojoj učenik mora napisati kod u c++ jeziku za ispis kubne vrijednosti svih brojeva između -5 i 5 , i kako bi to izgledalo u CPP tutoru.

Stručni modul CPP-tutor

Kako bi CPP-Tutor pružio inteligentnu povratnu informaciju studentu, Stručni modul se oslanja na grupu informacija: izjavu o problemu, specifikaciju problema, kod učenika, uspostavljeni studentski model, C++ kompilaciju i rezultat C++ runtime-a. Na temelju ovog konteksta, neke od ovih informacija neće biti dostupne. Međutim, cilj ovog modula je pažljivo analizirati sve dostupne informacije kako bi se mogla generirati odgovarajuća povratna informacija za učenika. To se postiže temeljnom komponentom Modula za prepoznavanje stručnih namjera.

Modul za prepoznavanje namjere stručnjaka

Svrha modula za prepoznavanje namjere stručnjaka (EIR – Expert intent recognition) je osigurati najvjerođostojniju predaju koda koju je učenik namjeravao. EIR se poziva kada standardni C++ prevoditelj zakaže. Pretpostavimo da se predano rješenje studenta zove Sol1, a postojeće rješenje za navedeni problem je Sol2. Modul EIR izvodi usklađivanje uzorka između dva rješenja (Sol1 i Sol2) za generiranje niza funkcije transformacije i za izračunavanje udaljenosti za uređivanje između studentovog rješenja i stvarnog rješenja pomoću algoritma za dinamičko programiranje (DPA). Modul EIR konstruira niz funkcija preslikavanja transformacije (tj. $T: Sol1 \rightarrow Sol2$). T uključuje sva umetanja, brisanja, transpozicije i promjene znakova koje su potrebne za transformaciju Sol1 u Sol2. EIR modul prosljeđuje ove informacije stručnom modulu kako bi odredio odgovarajući odgovor za studenta.



Slika 6. Dijagram toka za CPP tutor. (Izvor: Abu-Naser, 2008).

Slika 6 prikazuje dijagram toka za CPP tutor. On predstavlja sljedeće: nakon što student upiše kod, šalje ga kompajleru za c++ programski jezik, ako kod prođe kompajliranje odnosno učenik nije napravio nikakve sintaksne greške i točno je izvršio zadatak iz prve, u tom slučaju dobiva pozitivnu povratnu informaciju od sustava. Ako je učenik napravio neku grešku, sustav proba izvršiti napisani krivi kod i šalje učeniku negativnu povratnu informaciju, ako učenik popravi grešku ili mu sustav riješi problem, dobiva novi zadatak ili izlazi iz sustava, ako ne, tok dijagrama se završava i vjerojatno bi krenuo ispočetka

Prije nego što modul EIR izvrši bilo kakve promjene, studentu se postavljaju određena pitanja u svrhu pojašnjenja namjere.

Dakle, koristeći primjer na slici 5 , pojavio bi se sljedeći dijalog između CPP-tutora i učenika kao na slici 7

```

1. #include<iostream.h>
2. void main( {
3. Int c,j=-5
4. while(j>=5){
5. c:=j*j*j; j++; cout <<c<<<endl; })

```

CPP-Tutor: "It seems something is missing in line 2. It might be ')' between '(' and ')' Do you agree?"

Student: Yes

CPP-Tutor makes the correction and proceeds with rest of the code.

```

1. #include<iostream.h>
2. void main() {
3. Int c,j=-5
4. while(j>=5){
5. c:=j*j*j; j++; cout <<c<<<endl; })

```

CPP-Tutor: "I suppose that you need the keyword 'int' instead of 'Int'. Is that what you meant?"

Student: Yes

CPP-Tutor makes the correction and proceeds with rest of the code.

```

1. #include<iostream.h>
2. void main() {
3. int c,j=-5
4. while(j>=5){
5. c:=j*j*j; j++; cout <<c<<<endl; })

```

CPP-Tutor: "It seems something is missing in line 3, It might be ';' Do you agree?"

Student: Yes

CPP-Tutor makes the correction and proceeds with rest of the code.

```

1. #include<iostream.h>
2. void main() {
3. int c,j=-5;
4. while(j>=5){
5. c:=j*j*j; j++; cout <<c<<<endl; })

```

CPP-Tutor: "The symbol '>=' found in line 4, It should be '<=' instead'. Is that what you meant?"

Student: Yes

CPP-Tutor makes the correction and proceeds with rest of the code.

```

1. #include<iostream.h>
2. void main() {
3. int c,j=-5;
4. while(j<=5){
5. c:=j*j*j; j++; cout <<c<<<endl; })

```

CPP-Tutor: " The symbol ':=' found in line 5, It should be '=' instead'. You are learning C++ not Pascal Language. Is that what you meant?"

Student: Yes

Slika 7. Primjer uključivanja CPP tutora u dijalog s učenikom. (Izvor: Abu-Naser, 2008).

Proces podučavanja je dinamičan. Student se u bilo kojem trenutku može ne složiti s prijedlozima CPP-Tutora i modificirati izvorni kod. Nakon izmjene korisnika, proces podučavanja počinje ponovno. Vidimo kako bi izgledalo dijalog između tutora i učenika u stvarnom okruženju učenja u CPP tutoru, kako sustav daje pomoć i savjete kako bi učenik lakše otkrio gdje i zašto je problem u sintaksi.

Modul za povratne informacije CPP-tutora

Učinkovitost bilo kojeg inteligentnog sustava podučavanja uvelike ovisi o vremenu i stilu povratnih informacija. Vrijeme se odnosi na to kada učenik dobije odgovor na rješenje. Kada se povratna informacija prezentira učeniku, treba se rukovoditi onim što učenik zna. Na primjer, na početku poučavanja programiranja, mentor će možda htjeti ispraviti sve sintaktičke pogreške koje učenik čini. Kako učenik napreduje i uči više, povratne informacije o sintaksi jezika bile bi dosadne. Umjesto toga, sustav bi trebao predstaviti neposrednu povratnu informaciju za koncepte više razine koje učenik pokušava naučiti i odgoditi povratnu informaciju za koncepte niže razine.

Korisničko sučelje CPP-tutora i njegovo funkcioniranje.

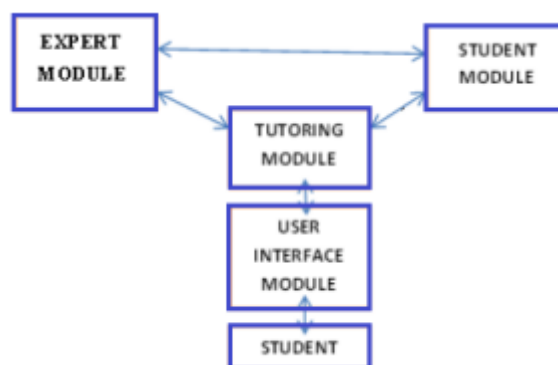
Korisničko sučelje temelji se na prezentacijskom formatu implementiranom u mnogim popularnim integriranim razvojnim okruženjima koja koriste profesionalni programeri (Abu-Naser, 2008; Conlan i dr ., 2002). Nakon spajanja na web stranicu CPP-Tutor, učenikov preglednik prikazuje radno okruženje za CPP-Tutor. Odabire se odgovarajući problem ili se učeniku prezentira problem koji je zadnji pokušao riješiti. U bilo kojem trenutku učenik može zatražiti novi zadatak. Učenik upisuje svoje rješenje u prozor rješenja. Nakon što učenik završi s odgovorom na problem, može odabrati gumb *Kompajliraj*. Nakon pritiska gumba *Kompajliraj*, učenikov kod se šalje C++ prevoditelju. Ako studentovo rješenje prođe fazu kompajliranja, C++ engine pokreće učenikov kod i izlaz se prikazuje u izlaznom prozoru. Ako faza kompajliranja ne uspije, poziva se postupak usklađivanja uzorka kako bi se izračunala udaljenost između učenikovog rješenja i stvarnog rješenja, predlaže se modificirano rješenje učenikovog koda, a

prikupljene informacije šalju se ekspertnom modulu radi pravilne povratne informacije. Kada učenik postane iskusniji programer, on ili ona može preskočiti poduku o određenom problemu pritiskom na gumb za pokretanje umjesto gumba za kompajliranje. Rezultat se prikazuje izravno u izlaznom prozoru. Učenik, u bilo kojem trenutku, može izričito zatražiti od CPP-tutora da vidi rješenje, izađe iz trenutnog problema i odabere novi i pregleda svoju izvedbu na temelju statistike uključujući probleme koji su probani riješeni, problemi koji su riješeni, broj pokušaja i poteškoće određenog problema (Abu-Naser, 2008).

3.6 Inteligentni tutorski sustav za C# programski jezik

Teorijskom analizom saznaje se prema Al-Bastami i Abu-Naser (2017) da je ovaj inteligentni tutorski sustav razvijen je korištenjem alata za izradu inteligentnih tutorskih sustava (ITSB) kako bi mogao pomoći studentima naučiti učinkovito programirati u C# i učinite postupak učenja vrlo učinkovitim. Korištena je baza znanja koja koristi ITSB stil autorskog alata. Studentovu radu daje individualne povratne informacije i podršku.

Arhitektura inteligentnog tutorskog sustava za poučavanje C# programskog jezika:



Slika 8 Arhitektura ITS-a za učenje C# programskog jezika. (Izvor: Al-Bastami & Abu-Naser, 2017).

Na slici 8 jasno se može vidjeti da je arhitektura inteligentnog tutorskog sustava za učenje c# programskog jezika izrađena od 4 glavna modula od kojih se obično neki inteligentni tutorski sustav sastoji ili barem sadržava u svojoj strukturi.

Studentski modul

Prije nego što novi student može koristiti C# inteligentni tutorski sustav, on/ona mora imati profil u sustavu. Profil sadrži informacije o studentu kao što su: datum zadnjeg predavanja, ime studenta, broj studenta, trenutni rezultat, ukupni rezultat, završena razina težine za svaku lekciju i broj problema tijekom svake sesije. Trenutni rezultat predstavlja rezultat učenika za trenutnu razinu. Ukupni rezultat predstavlja rezultat učenika za sve razine.

Tutorski modul

Tutorski modul radi kao koordinator koji kontrolira funkcionalnost inteligentnog tutorskog sustava jezika C#. Putem ovog modela učenik može odgovoriti na generirana pitanja u svakoj razini svake lekcije i ako učenik dobije više od 70%, on/ona može prijeći na drugu razinu. Ali ako on/ona dobije ocjenu između 50% i manje od 70% zatim ponavlja vježbe iste razine; ako on/ona dobije ocjenu manju od 50%, on/ona će biti vraćen na lekciju da je dobro prouči, a zatim će se vratiti opet isprobati vježbe.

Korisničko sučelje

ITSB autorski alat za izradu inteligentnih tutorskih sustava, koji se koristi za izgradnju inteligentnog tutorskog sustava za C# jezik podržava korisničko sučelje i sučelje nastavnika. Kada je nastavnikov dnevnik u sustavu, on/ona može dodati: podaci o studentu, lekcije, vježbe, odgovori, fontove, nazive i veličine svih gumba, izbornika, kombinirane konfiguracije, boju pozadine i boju teksta za sve obrasce u sustavu itd. Dakle, ovo sučelje pruža sustav s potrebnom fleksibilnošću za korištenje od strane nastavnika i studenta.

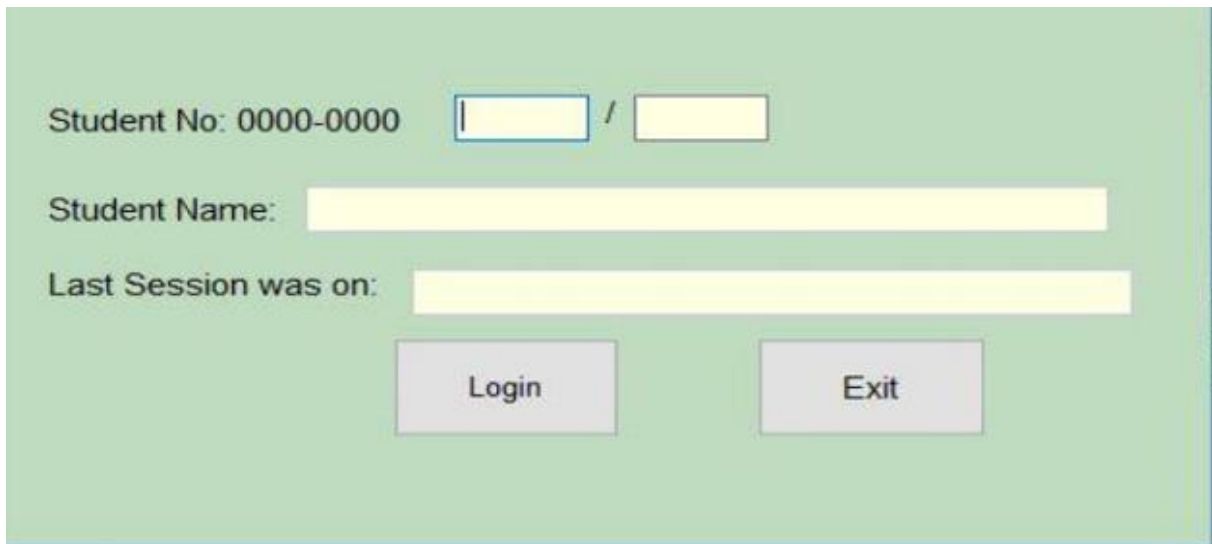
Ekspertni modul

Ova komponenta predstavlja modul domene. Sadrži znanje iz lekcija koje treba podučavati učenicima u Inteligentnom tutorskom sustavu za podučavanje C# programskog jezika. Lekcije su sljedeće:

- Uvod u programiranje
- Tipovi podataka i varijable
- Operatori i izrazi

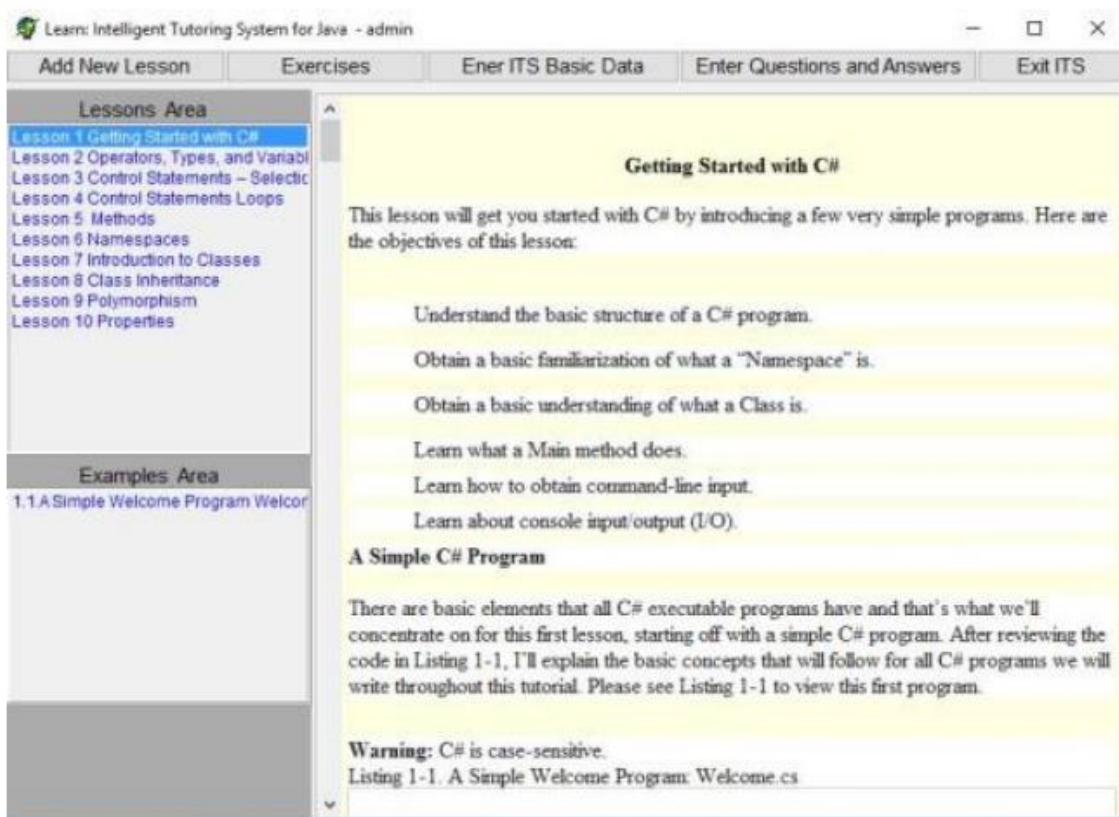
- Ulaz i izlaz konzole
- Uvjeti
- Petlje
- nizovi
- Brojevnii sustavi
- Metode
- Rekurzija
- Stvaranje i korištenje objekata
- Obrada nizova i teksta
- Definiranje klasa
- Tekstualne datoteke
- Linearne strukture podataka
- Stabla i grafikoni
- Rječnici, hash-tablice i setovi

U nastavku redom slijede slike korisničkog sučelja i primjeri korištenja Inteligentnog tutorskog sustava za učenje C# programskog jezika: login screen, biranje lekcije koju korisnik želi učiti, biranje primjera iz te lekcije koju želi učiti, pitanja za korisnika nakon čitanja lekcije i evaluacija odgovora, dodavanje novog studenta i upis podataka o njemu, sučelje o statusu, razini pitanja, problemi i bodovi, pop up prozora ukoliko korisnik ne uspije odgovoriti na određeni potrebni postotak pitanja točno s porukom u kojoj mu se preporuča da se vrati na prethodnu razinu iz te lekcije i pokuša ponovo (Al-Bastami, Abu-Naser, 2017).



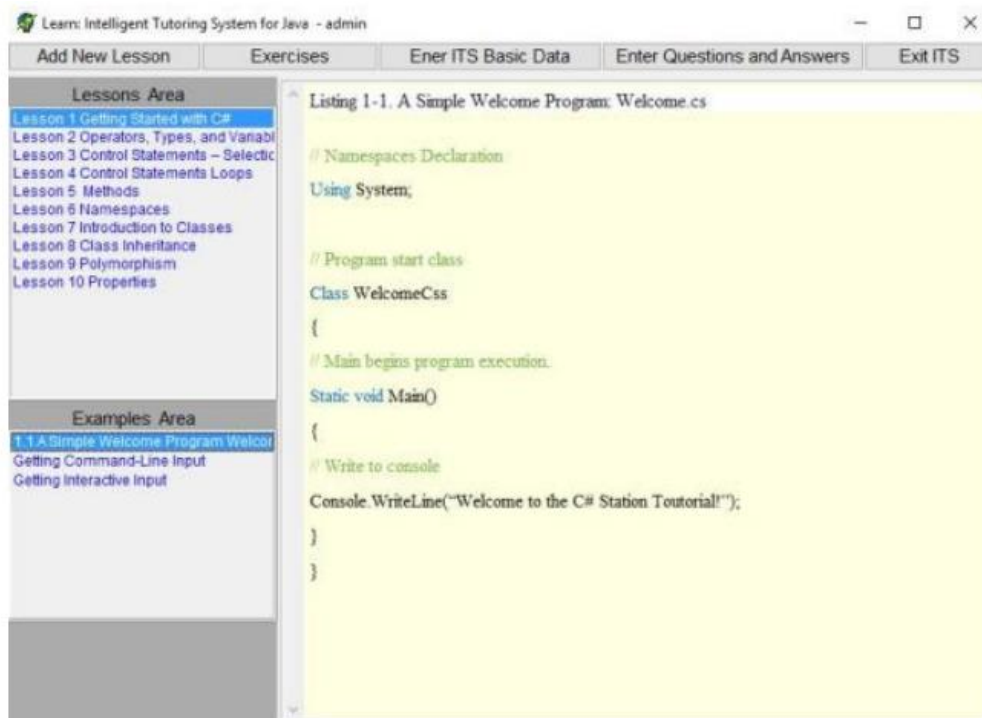
Slika 9. Login screen Inteligentnog tutorskog sustava za učenje C# programskog jezika. (Izvor: Al-Bastami & Abu-Naser, 2017).

Na login stranici učenik upisuje svoj broj, ime i prezime te vrijeme kad je bila zadnja sesija na kojoj je sudjelovao nakon toga pritišče prijavi se ili izlaz.



Slika 10. Biranje lekcije koju učenik želi učiti iz Inteligentnog tutorskog sustava za učenje C# programskog jezika. (Izvor: Al-Bastami & Abu-Naser, 2017).

Slika 10 prikazuje izgled sučelja Inteligentnog tutorskog sustava za učenje C# programskog jezika na kojoj učenik može birati lekcije i primjere iz te lekcije koju odabere za učenje te dio sadržaja iz lekcije 1.



Slika 11. Biranje primjera iz lekcije koju učenik želi učiti iz Inteligentni tutorski sustav za učenje C# programskog jezika. (Izvor: Al-Bastami & Abu-Naser, 2017).

Slika 11 prikazuje izgled sučelja Inteligentnog tutorskog sustava za učenje C# programskog jezika na kojoj učenik može birati lekcije i primjere iz te lekcije koju odabere za učenje te dio sadržaja iz primjera lekcije 1.



Slika 12. Pitanja za korisnika nakon čitanja lekcije iz Inteligentni tutorski sustav za učenje C# programskog jezika. (Izvor: Al-Bastami & Abu-Naser, 2017).

Slika 12 prikazuje izgled sučelja Inteligentnog tutorskog sustava za učenje C# programskog jezika na kojoj učenik odgovara na pitanja koja mu sustav postavlja nakon čitanja lekcije, konkretno u ovom slučaju pitanje iz prve lekcije problem #3, težina učenja #1. Pitanja su višestrukog odabira i učenik u svakom trenutku može provjeriti svoj odgovor, zatražiti rješenje, pogledati statistiku ili izaći.

1. Correct answer:

Excellent, The Answer is Correct

2. Wrong answer:

Sorry, The Answer is Wrong

Slika 13. Evaluacija odgovora nakon pitanja za korisnika nakon čitanja lekcije iz Inteligentni tutorski sustav za učenje C# programskog jezika. (Izvor: Al-Bastami & Abu-Naser, 2017).

Slika 13 prikazuje kako Inteligentni tutorski sustav za učenje C# programskog jezika prikazuje evaluaciju odgovora koji je student označio i odlučio provjeriti.

Constants Data Entry

ITS Basic Data Students Data Colors

Enter Student Number

Enter Student Name

Enter Student Major

Enter Student Grade Point Average

Enter Student Passed Credits

Re-Set Student Difficulty Level

Re-Set Student Problem No

Re-Set Student Current Score

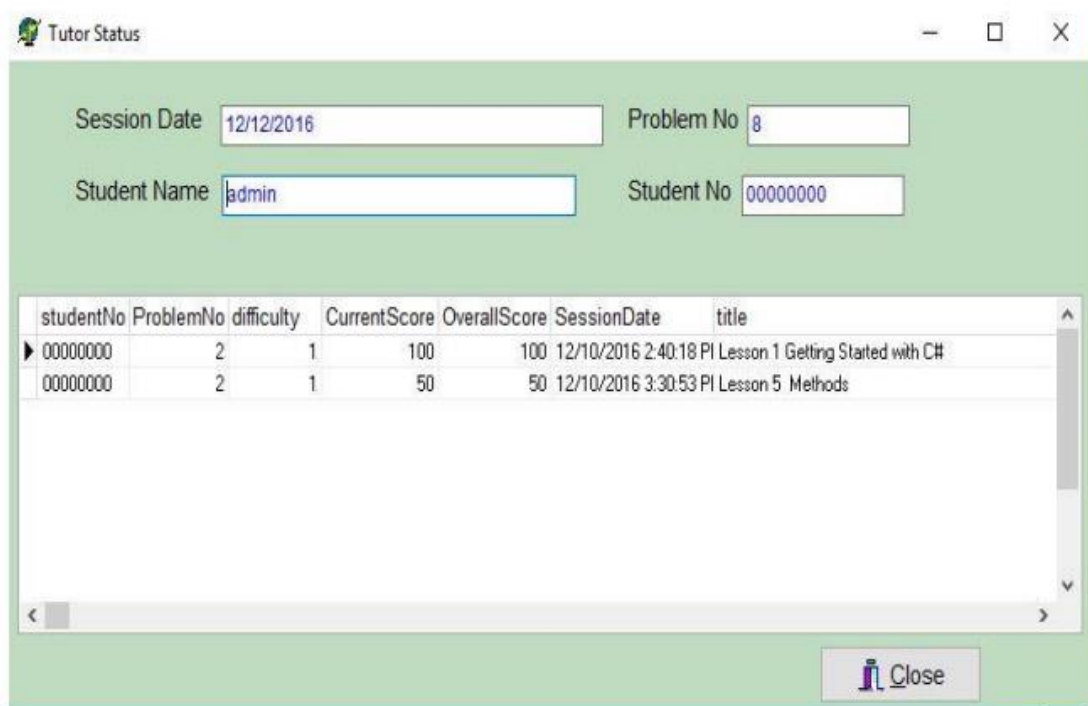
Re-Set Student Over All Score

Re-Set Student Current Lesson

Navigation buttons: [Back], [Previous], [Next], [Forward], [Search], [Home], [Refresh], [Close]

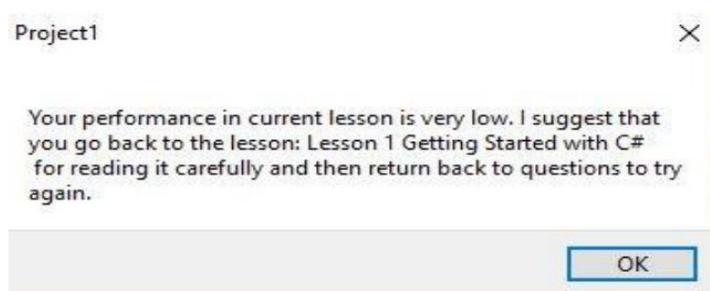
Slika 14. Primjer dodavanja novog studenta i upis podataka o njemu iz Inteligentni tutorski sustav za učenje C# programskog jezika. (Izvor: Al-Bastami & Abu-Naser, 2017).

Slika 14 prikazuje stranicu sučelja Inteligentnog tutorskog sustava za učenje C# programskog jezika na kojoj se dodaje novi student, i upisuju podaci o studentu, broj studenta, ime, njegov prosjek ocjena, razina težine učenja, može se ponovno postaviti njegov trenutni broj bodova, broj lekcija i problema koje je riješio, te ukupni broj bodova koje je student osvojio na 0.



Slika 15. Primjer izgleda sučelja o statusu, razini pitanja, problema i ostvarenih bodova iz Inteligentni tutorski sustav za učenje C# programskog jezika. (Izvor: Al-Bastami & Abu-Naser, 2017).

Slika 15 prikazuje stranicu sučelja Inteligentnog tutorskog sustava za učenje C# programskog jezika na kojoj učenik može vidjeti podatke o statusu i broju bodova u odnosu na prosječni broj bodova koji se osvoji na tom problemu unutar neke lekcije prema datumu sesije, imenu, problemu na kojem se nalazi unutar neke lekcije, broju učenika.



Slika 16. Primjer izgleda pop up prozora sa savjetom tutora iz Inteligentni tutorski sustav za učenje C# programskog jezika. (Izvor: Al-Bastami & Abu-Naser, 2017).

Slika 16 prikazuje stranicu sučelja Inteligentnog tutorskog sustava za učenje C# programskog jezika gdje sustav izbacuje pop-up obavijest u slučaju da učenik netočno

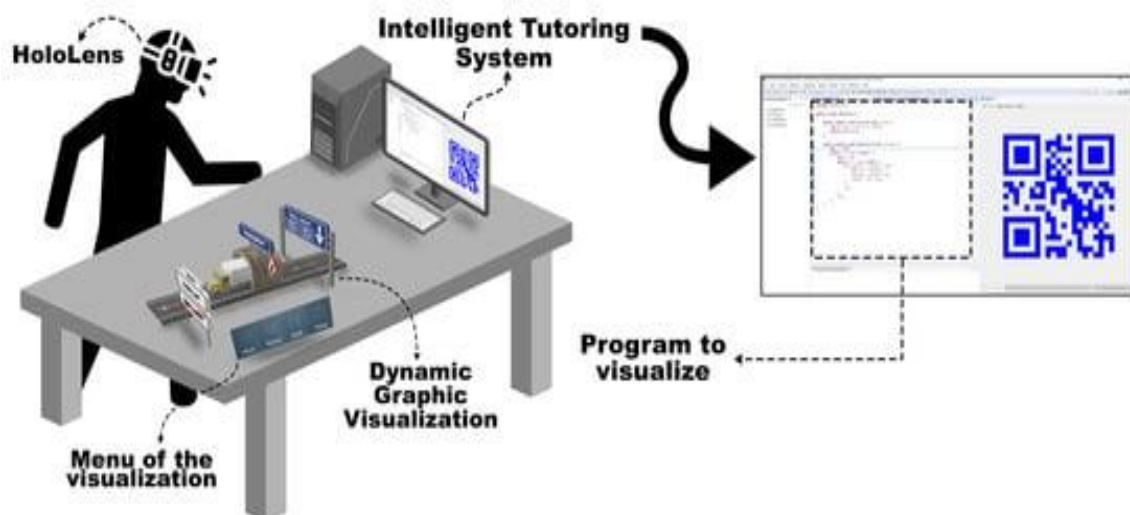
odgovori na određeni potrebni postotak pitanja točno, uz poruku u kojoj mu se preporuča da se vrati na prethodnu razinu i pokuša ponovo riješiti prvo prethodnu lekciju i onda neka se pokuša vratiti na ovu lekciju na kojoj nije odgovorio točno na određeni potrebni broj pitanja točno.

3.7 Inteligentni tutorski sustav za poučavanje programiranja uz korištenje dinamičkih grafičkih vizualizacija

Poseban slučaj u kojem umjetna inteligencija ima značajnu važnost je obrazovanje, čiji je jedan od prioritarnih ciljeva eliminacija tradicionalnog obrazovnog modela (Mu, 2019).

U ovom pristupu može se dodati korištenje grafičkih vizualizacija koji povezuju koncepte programiranja s drugim konceptima s kojima bi učenici mogli biti bolje upoznati.

Kako bi se poboljšalo razumijevanje učenja programiranja, Schez-Sobrino i dr. (2020) uspostavili su ITS koji pomaže u programiranju pomoću autonomnih dinamičkih vizualizacija koje vode učenika tijekom izvođenja programa. Opća shema sustava prikazana je na slici 16.



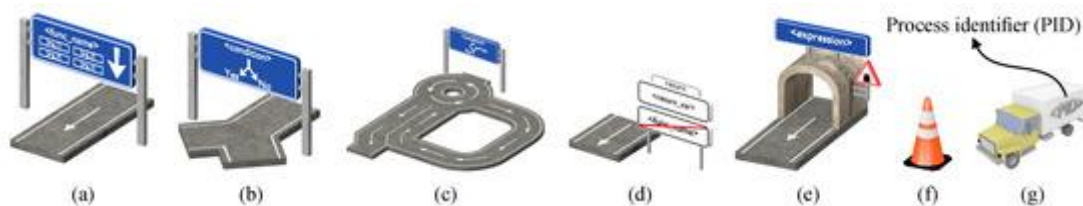
Slika 17. Opći pregled predloženog inteligentnog tutorskog sustava. Izvor: Schez-Sobrinoi dr., 2020).

Inteligentni tutorski sustav za poučavanje programiranja uz korištenje dinamičkih grafičkih vizualizacija ANGELA

Sustav ANGELA ima za cilj olakšati uvođenje programiranja studentima koji tek počinju programirati i nemaju dovoljno znanja za rješavanje problema pomoću nekog programskog jezika. Uglavnom je situacija povezana s apstraktnošću, potrebom za razumijevanje koncepata programiranja koji se podučavaju, s nedostatkom znanja jezika ili s poteškoćama u dizajniranju i izradi algoritama (Schez-Sobrinu i dr., 2020. prema Bosse, Gerosa, 2017).

Sustav integrira korištenje metafora koje omogućuju smanjenje ove razine apstrakcije uključivanjem analogija između koncepata iz stvarnog svijeta i aspekata programiranja. Ova analogija uspostavlja notaciju s cestama i prometnim znakovima.

Slika 18 prikazuje skup elemenata koji su dio notacije:



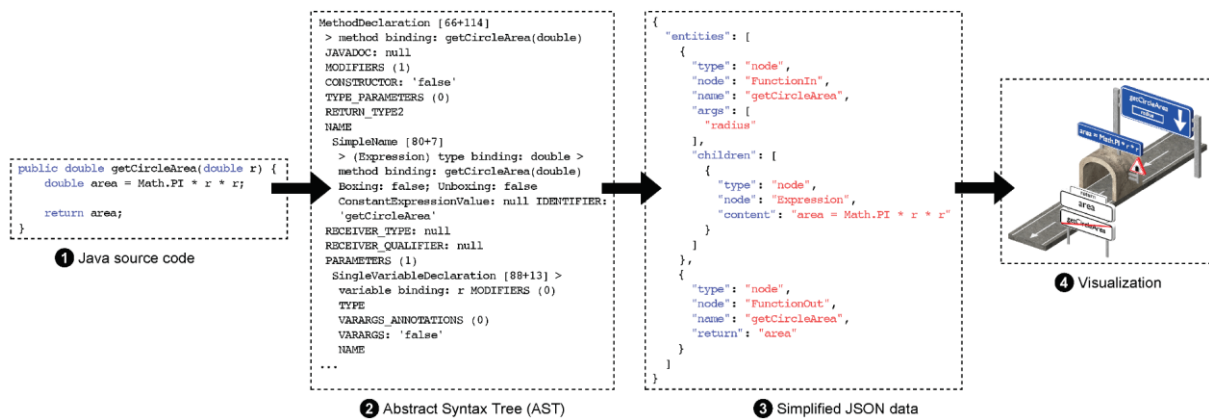
Slika 18. Grafički prikazi notacije prometnih znakova za olakšavanje učenja programiranja pretvoreni u ortogonalno projicirani 3D prostor. (Izvor: Schez-Sobrinu i dr., 2020).

Slika 18: (a) Definicija funkcije. (b) Uvjetna izjava. (c) petlje. (d) Return funkcije. (e) Procjena izraza. (f) Prijelomna točka. (g) Programska nit.

Sustav je olakšao izradu vizualizacija automatskim generiranjem iz izvornog koda programa. Generirane vizualizacije zatim su poslone AR uređaju koji je djelovao kao preglednik koji je putem mreže primao potrebne informacije o vizualizaciji od Inteligentnog tutorskog sustava.

Za to je iz izvornog koda programa generirana pojednostavljena hijerarhija u JavaScript Object Notation (JSON (format za razmjenu podataka koji koristi tekst za prijenos podataka koji se sastoji od para atribut-vrijednost)) Kako bi to učinio, Inteligentni tutorski sustav je analizirao apstraktno sintaksno stablo (AST) generirano iz izvornog koda i identificirao svaki od jezičnih elemenata povezanih s predloženim

grafičkim prikazima (slika 18). Hijerarhija generirana u JSON formatu zadržala je svaku od ovih asocijacija u obliku stabla, zajedno s konektorima potrebnim za konstruiranje vizualizacije na povezan način.

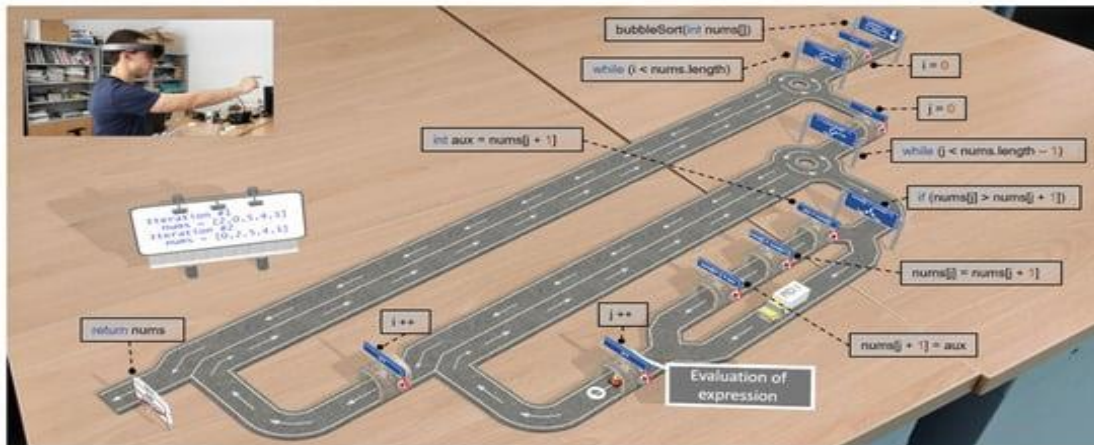


Slika 19. Generiranje JSON strukture s relevantnim informacijama koje se koriste za izgradnju vizualizacije iz izvornog koda programa. (Izvor: Schez-Sobrinio i dr., 2020).

Slika 19 prikazuje primjer pojednostavljenih pothijerarhiji generiranih za neke od predloženih grafičkih prikaza. Konačna vizualizacija bi sadržavala kompletnu hijerarhiju iz ovih pothijerarhiji: (1) izvorni kod programskog jezika Java, (2) AST, (3) pojednostavljeni JSON podaci, (4) vizualizacija.

Trenutačno se implementacija Inteligentnog tutorskog sustava provodi putem plug-ina implementiranog za COLLECE-2.0, kolaborativni distribuirani sustav za učenje programiranja temeljen na razvojnom okruženju Eclipse. Ovaj dodatak koristi infrastrukturu COLLECE-2.0 za generiranje JSON-a iz AST-a izvornog koda programa. Program za ispravljanje pogrešaka koristi Inteligentni tutorski sustav putem API-ja (Aplikacijsko programsko sučelje) koji pruža JDT Debug plug-in iz Eclipsea.

Slika 20 na sljedećoj stranici prikazuje potpuni primjer dinamičke vizualizacije generirane pomoću predloženog ITS-a putem AR uređaja (Schez-Sobrinio i dr., 2020).



Slika 20. Dinamička vizualizacija algoritma “Bubble sort” viđena kroz AR uređaj. (Izvor: Schez-Sobrinio i dr., 2020).

4. PREDNOSTI I OGRANIČENJA INTELIGENTNIH TUTORSKIH SUSTAVA ZA POUČAVANJE PROGRAMIRANJA

Teorijskom analizom inteligentnih tutorskih sustava izdvojile su se neke od prednosti i ograničenja inteligentnih tutorskih sustava.

Prednosti:

- Dugoročno smanjuje troškove
- Optimiziraju rad nastavnika i njegov razvoj
 - Neke vrste testova mogu se implementirati u Inteligentni tutorski sustav, tako bi se oslobodilo nastavnika tih „rutinskih“ zadataka i stvorilo više vremena za učinkovitije korištenje istog i podizanje učenja na višu razinu
- simuliraju individualno podučavanje, daju ciljane pravovremene povratne informacije i grade personalizirane putove učenja kako bi najbolje zadovoljili individualne potrebe učenika za učenjem. Mogu učenje učiniti personaliziranijim, fleksibilnijim i zanimljivijim.
- Dostupnost
 - Većina nastavnika i učenika preferira Inteligentni tutorski sustav zbog njegove dostupnosti. Učenici mogu koristiti ove sustave za učenje u bilo koje doba dana s bilo koje lokacije. Dok nastavnik u učionici ima ograničeno vrijeme podučavanja, Inteligentni tutorski sustav je uvijek dostupan i omogućuje nesmetan proces učenja (Sharma, Harkishan 2022).
- Pomažu u optimizaciji putova učenja i razvoju stabla odlučivanja. Kada su primjene umjetne inteligencije u učenju bile prilično ograničene, "adaptivno učenje" značilo je da učenik slijedi jedan od unaprijed zadanih putova učenja (nastavnikov unaprijed izrađeni plan nastave za tu godinu). Nasuprot tome, moderni Inteligentni tutorski sustav nema unaprijed postavljene staze učenja. Uz AI modeliranje, "nova prilagodljivost" znači prilagođavanje izvedbi svakog učenika i optimiziranje strategija podučavanja (Valverde, 2020).
- Veći angažman učenika
 - Umjesto pasivnog načina poučavanja, institucije bi mogle personalizirati okruženja za učenje za bolji angažman.
- Omogućuje učenje vlastitim tempom

- Povećava motivaciju
- Poboljšava ICT vještine, te učenik bolje razumije ICT i gradivo
- Infografika.
 - Učenicima se prenosi nesaglediva količina informacija, a razumijevanje se pokazalo neuspjelim. Infografika je jasan, umjetnički i koncizan način prezentiranja informacija na jedinstven način. Učenicima postaje sve teže snalaziti se kroz duge bilješke i materijale za obuku. Međutim, Inteligentni tutorski sustav nudi infografske sustave koji učenicima omogućuju brzo usvajanje informacija.
- Mehanizam povratne informacije
 - ITS daje povratnu informaciju studentima i mentoru što pomaže u poboljšanju učinka i maksimiziranju vremena. Neki Inteligentni tutorski sustavi mogu istaknuti pogrešne odgovore dok rješavaju različita pitanja kako bi vodili učenike. Također može dati trenutni savjet kada učenik podnese zahtjev ... to učenike motivira da naprave određene promjene da poboljšaju svoju izvedbu te također dublje promišljaju o svojoj izvedbi.
- Omogućava bolje upravljanje vremenom
 - Kada se vremenom ispravno upravlja, produktivnost bi bila poboljšana i za instruktora i za učenika. Inteligentni sustavi podučavanja koriste napredne algoritme za automatsko dodjeljivanje rasporeda obuke i učenja za učenike i nastavnike, što znači da učenje može biti učinkovitije (Akyuz, 2020).
- Nepristrana prosudba
 - ITS analizira i ocjenjuje učenike na temelju njihovih vještina, osobina i uspješnosti što smanjuje šanse za pristrane prosudbe i nepravedan tretman (Akyuz, 2020, prema Lynch, 2018).

Ograničenja

- Društvena otuđenost
 - Opsežnom upotrebom tehnoloških rješenja učenici bi bili sve više izolirani od svojih prijatelja i učitelja. Ljudi su društvena bića i fizička interakcija s ljudima ima veliki značaj u ljudskom razvoju.

- Izlaže učenika neželjenim sadržajima ili čak štetnih ako je na internetu. (npr. pornografija ili nasilje na internetu)
- Smanjuje komunikacijske vještine
 - Svakodnevna interakcija s ljudima poboljšava ljudske komunikacijske vještine, ali korištenje tehnologije za učenje može uvelike spriječiti ljude u međusobnoj komunikaciji (Akyuz, Y., 2020).
- Ovisnost o tehnologiji
 - Tehnologija može uvelike spriječiti praktične vještine i fizičke angažmane. Povećano oslanjanje na tehnologiju utjecat će na sposobnost ljudi da se predstave u stvarnim životnim situacijama i samo će stvoriti društveno neugodne odrasle osobe ovisnike o tehnologiji (Lynch, 2018).
- Inteligentni tutorski sustavi skupi su i za razvoj i za implementaciju.
 - Na primjer, ankete su otkrile da je za kodiranje jednog sata online podučavanja potrebno 300 sati razvoja sadržaja podučavanja (Murray T., 1999) .

5. BUDUĆI RAZVOJ INTELIGENTNIH TUTORSKIH SUSTAVA ZA POUČAVANJE PROGRAMIRANJA

Kako se tehnologija svakim danom nastavlja razvijati, tako raste i potencijal implementacije Inteligentnih tutorskih sustava u obrazovanje. Inteligentni tutorski sustavi do sad su pokazali svoju praktičnost i učinkovitost, no njihova izgradnja teška je i naporna. Brojni su izgrađeni, no još uvijek nisu premostili potrebu za stvarnim ljudskim učiteljem, pristupačnost, te samim tim onda i uspješnost. Izazovi i područja istraživanja na koja se treba usredotočiti nabrojani su u nastavku kao prijedlozi za poboljšanja dosadašnjih izgrađenih Inteligentnih tutorskih sustava koji su postali jasni teorijskom analizom Inteligentnih tutorskih sustava tijekom pisanja ovog rada kako bi postigli svoj puni potencijal.

PRIJEDLOZI ZA POBOLJŠANJE

- **Težine za izgradnju**

Razvoj novog Inteligentnog tutorskog sustava težak je i zahtijeva složenost u razmišljanju i suradnji između stručnjaka za domenu i razvojnih programera kojima treba 200-300 sati razvoja za samo jedan sat nastave u Inteligentnom tutorskom sustavu. Stoga postoji snažan interes za pojednostavljenjem procesa izgradnje jer bi se tako stvaranje novog Inteligentnog tutorskog sustava omogućio lakšim i pristupačnijim za nastavnike i programere. Postoji nekoliko tehnika i autorskih alata koji bi ovaj proces olakšali i smanjili napor izgradnje ali glavna kategorija u koju bi mogli podijeliti ih je oni koji zahtijevaju i oni koji ne zahtijevaju vještine programiranja. Istraživači Ken R. Koedinger Daniel Weitekamp i Erik Harpstead sa sveučilišta Carnegie Mellon pokazali su da ih se može brzo izgraditi tako da nauče računalo da podučava. Učitelj poučava računalo pokazujući mu načine na koji se problem u temi rješava. Značajno je da računalo ne nauči samo riješiti problem nego i načine na koje se problem rješava, generalizirati kako bi znao riješiti i ostale nadolazeće probleme u temi. Objavili su rad koji opisuje metodu izrade koja koristi program strojnog učenja koji simulira učenikovo učenje. Metoda ubrzava razvoj Inteligentnih

tutorskih sustava i omogućuje učiteljima da sami stvore računalne lekcije bez potrebe za razvojnim programerom. Weitekamp je razvio nastavno sučelje za mašinu strojnog učenja koja koristi metodu pokaži i ispravi, koja ne zahtijeva programiranje i puno je lakša. Predložena metoda bi mogla omogućiti izradu 30-minutne lekcije za 30 minuta. Koedinger je to nazvao velikom vizijom među razvojnim programerima. Učiteljima to također omogućava i primjenu vlastite preferencije poučavanja nekog gradiva. Također ova metoda mogla bi dovesti do dubljeg prepoznavanja težine gradiva za učenike s kojima se oni kao stručnjaci ne susreću jer bi računalo zapelo u učenju kao i učenik na težim dijelovima gradiva. Njihovo istraživanje djelomično je bilo podržano od Google-a i instituta za odgojne znanosti (Weitekamp et al., 2020).

- **Interaktivnosti**

Ljudski učitelji imaju sposobnost razumjeti nečiji ton unutar dijaloga i protumačiti to kako bi pružili kontinuiranu povratnu informaciju kroz kontinuirani dijalog. Sada se razvijaju inteligentni sustavi koji pokušavaju simulirati prirodne razgovore. Da bi dobili potpuno iskustvo dijaloga, postoji mnogo različitih područja u kojima se računalo mora programirati, uključujući sposobnost razumijevanja tona, govora tijela i izraza lica i zatim odgovaranja na njih. Dijalog u Inteligentnom tutorskom sustavu može se koristiti za postavljanje specifičnih pitanja kako bi se pomoglo usmjeravanju učenika i izvlačenju informacija, dok se učenicima omogućuje da izgrade vlastito znanje (Graesser et al., 2005).

- **Uvođenje gamifikacije i XR**

- Spomenut je ranije u radu predložen sustav ANGELA, Inteligentni tutorski sustav za poučavanje programiranja uz pomoć dinamičke vizualizacije.

Sučelja i mogućnost vizualizacije vjerojatno će nastaviti napredovati. Budućnost podučavanja uz pomoć umjetne inteligencije mogla bi se pretvoriti i u podučavanje kroz proširene stvarnosti. Na primjer XR,

skupni pojam koji pokriva i virtualnu stvarnost (VR), proširenu stvarnost (AR) i mješovitu stvarnost (MR) (21K School, 2023).

- **Rad na interpretaciji afektivnih procesa, motivacije i emocija**

Ljudi ne koriste samo kognitivne procese u učenju, već i afektivni procesi kroz koje prolaze također igraju važnu ulogu. Na primjer, učenici bolje uče kada imaju određenu razinu neravnoteže (frustracije), ali nedovoljno da bi se učenik osjećao potpuno preplavljenim. To je motiviralo afektivno računalstvo da počne proizvoditi i istraživati stvaranje inteligentnih sustava podučavanja koji mogu interpretirati afektivni proces pojedinca. Inteligentni tutorski sustav se može razviti za čitanje izraza lica pojedinca i drugih znakova afekta u pokušaju pronalaženja i podučavanja optimalnom afektivnom stanju za učenje. Postoje mnoge komplikacije u tome jer se afekt ne izražava samo na jedan način, već na više načina, tako da bi Inteligentni tutorski sustav bio učinkovit u tumačenju afektivnih stanja može zahtijevati multi modalni pristup (ton, izraz lica, itd...) (D'Mello, Graessner, 2012). Ove su ideje stvorile novo područje unutar Inteligentnih tutorskih sustava, Sustav afektivnog podučavanja (ATS) (Sarrafzadeh et al., 2008).

Predloženo je da bi za učinkovito dizajniranje Inteligentnih tutorskih sustava koji gradi odnos s učenicima, Inteligentni tutorski sustav trebao oponašati strategije neposrednog podučavanja, ponašanja koja premošćuju prividnu društvenu distancu između učenika i nastavnika, kao što je osmjehivanje i oslovljavanje učenika imenom (Wang et al., 2009). Motivacija ima golem utjecaj na učinkovitost i ishode učenja učenika.

- **Obrada prirodnog jezika**

Osim korištenja naprednih grafičkih tehnika, NLP je ključna komponenta za poboljšanje interakcije čovjeka i računala.

- **Integracija napredne analitike**

Inteligentni tutorski sustav bi mogli pružiti dublji uvid o uspješnosti, obrascima učenja, predvidjeti ishode budućeg učenja (Valverde, 2020).

- **Više domena znanja**

Sierra i dr., 2007. godine u radu „A Multi- agent Intelligent Tutoring System for Learning Programming“ obradili su ovu tezu na primjeru ITS-a za poučavanje programiranja. Cilj sustava je bio poučavanje ciljnog područja

uz pomoć potpornog područja. Zbog implementacije morali su odabrati multi-agentski sustav zbog karakteristika agenta u Inteligentnom tutorskom sustavu, jer pružaju fleksibilnost za buduće širenje, na još domena (Sierra i dr., 2007).

U njihovom slučaju domene su bile iz dvije grane programiranja, no zamislimo koja revolucija Inteligentnih tutorskih sustava bi se desila da se izbjegne potreba prilikom izgradnje za modeliranje i izgradnja ostalih potrebnih modela Inteligentnih tutorskih sustava, nego da se postigne izgradnja globalnog web-based tutorskog sustava s mogućnošću odabira jezika učenja, s više agenata za pretragu i spajanje znanja iz više domena, mogućnost dodavanja domeni, i proširivanje znanja postojećim uvrštavanjem značajke opisane u prvom prijedlogu ovog odlomka, čak i iz samo jedne grane poučavanja, te značajke ove sustave učinile bi revolucionarnima.

Iako ovi prijedlozi za poboljšanjima predstavljaju veliki napredak za stvarnu primjenu Inteligentnih tutorskih sustava, izazivaju nove probleme u vezi s privatnošću učenika i etikom koja se mora riješiti prije široke implementacije u obrazovanje zbog zakona o umjetnoj inteligenciji.

6. ZAKLJUČAK

Nekoliko provedenih studija, dokazalo je već više puta višu učinkovitost i kvalitetu učenja korištenjem Inteligentnih tutorskih sustava. Instant-ne povratne informacije, pristup kao jedan na jedan, dostupnost 24/7, odgovaranje bez anksioznosti i pritiska, motivacija za učenjem, veća i viša iskoristivost nastavnikova vremena za pripremanje plana nastave sigurno je veliko potpomoglo tome. No implementacija Inteligentnih tutorskih sustava u obrazovne ustanove je za sada budućnost, jer 1 sat nastave u učionici zahtijeva stotine sati programiranja 1 sata nastave preko Inteligentnih tutorskih sustava. No, kada bi se implementirali, mogli bi iz temelja promijeniti način poučavanja koji poznajemo. Što se tiče učenja programiranja, Inteligentni tutorski sustav za učenje programiranja bi čak mogao prije zaživjeti nego u ostalim područjima jer već postoje kompajleri pa je najviše pažnje posvećeno upravo korisničkom sučelju a što i je najveći problem pri stvaranju istog, gdje je po meni korak ispred nego ostala područja. Što se tiče korištenja izvan obrazovanja u npr. osposobljavanja radnika, neke visoko stojeće firme to si mogu priuštiti i koriste. Potencijalni utjecaj tek treba vidjeti no osobno smatram da je jasno da je budućnost Inteligentnih tutorskih sustava svijetla.

Osobno najveća kritika na Inteligentni tutorske sustave koji se trenutno koriste je pedagogija neposredne povratne informacije i nizova savjeta koji su ugrađeni kako bi sustav učinili "inteligentnim". Ovu pedagogiju kritiziram zbog neuspjeha u razvijanju dubokog učenja kod učenika. Kada učenici dobiju kontrolu nad sposobnošću primanja savjeta, stvorena reakcija učenja je negativna. Neki se učenici odmah okreću savjetima prije nego pokušaju riješiti problem ili dovršiti zadatak. Izdaje nerealne rezultate znanja učenika na kraju tečaja.

SAŽETAK

Sve veći utjecaj tehnologije mijenja način na koji ljudi uče i kako institucije podučavaju. Inteligentni tutorski sustavi nisu neka nategnuta futuristička ideja. U praktičnom smislu, oni već postoje. Iako su neobični, sposobni su funkcionirati bez prisutnosti instruktora i mogu učinkovito izazvati i pomoći učeniku korištenjem različitih algoritama. Termini tutor i mentor vežu se za neku osobu koja daje stručnu pomoć u nekom određenom području, u svladavanju neke određene vještine, prenošenju znanja iz cijelog tog područja u procesu učenja. Ako se to prenese na kontekst Umjetne inteligencije i Informacijsko-komunikacijske tehnologije, inteligentni tutorski sustav (Intelligent Tutoring Systems, ITS) je softver koji simulira ponašanje tutora/mentora/učitelja/nastavnika ovisno o njegovoj implementaciji. S ovakvim sustavima podiže se efikasnost koja nije pronađena unatoč promicanju e-učenja i istraživanjima u koja su se uključili istraživači iz cijelog svijeta u nedavno proživljenoj zdravstvenoj krizi corona virusa (pandemije covid-19). U ovom radu opisuju se i analiziraju Inteligentni tutorski sustavi kao rješenje problema dostupnosti, efikasnosti, cjeloživotnog-visokog obrazovanja na teorijski način uz pomoć provedenih studija nekih od istraživača s ciljem promidžbe Inteligentnih tutorskih sustava i nekorištenja Inteligentnih tutorskih sustava u visokoškolskim ustanovama. Cilj je istražiti i opisati faze njihova razvoja i glavne modele od kojih se sačinjava arhitektura, kako se polazi nastava programiranja jednom kad se implementiraju na temelju studija nekih od istraživača, te utvrditi njihove prednosti i nedostatke. Spomenuli su se i opisale strukture i rad odnosno učenje u nekima od inteligentnih tutorskih sustava za programiranje. Predstavili su se također na primjeru inteligentni tutorski sustavi za poučavanje programiranja uz korištenje dinamičkih grafičkih vizualizacija. Predložio se budući razvoj s prijedlozima za poboljšanje dosadašnjih. Zaključilo se da je implementacija Inteligentnih tutorskih sustava u obrazovne ustanove za sada budućnost, jer 1 sat nastave u učionici zahtijeva stotine sati programiranja 1 sata nastave preko Inteligentnih tutorskih sustava. No, kada bi se implementirali, mogli bi iz temelja promijeniti način poučavanja koji poznajemo. Što se tiče učenja programiranja, inteligentni tutorski sustav za učenje programiranja bi čak mogao prije zaživjeti nego u ostalim područjima. Potencijalni utjecaj tek treba vidjeti no smatra se da je jasno da je budućnost Inteligentnih tutorskih sustava svijetla.

Ključne riječi: Inteligentni tutorski sustavi, programiranje, poučavanje

ABSTRACT

The growing impact of technology is changing the way people learn and how institutions teach. Intelligent tutoring systems are not some tense futuristic idea. In practical terms, they already exist. Although they're unusual, they are able to function without the presence of instructors and can effectively challenge and assist the student by using different algorithms. Terms tutor and mentor are tied to a person who provides professional assistance in a particular area, in mastering a particular skill, in transferring knowledge from all over the field in the learning process. If this is transferred to the context of Artificial Intelligence and Information and Communication Technology, the intelligent tutoring system (Intelligent Tutoring Systems, ITS) is software that simulates the behavior of tutors/mentors/teachers depending on its implementation. With such systems, efficiency that has not been found is raised despite the promotion of e-learning and research involving researchers from around the world in the recently experienced health crisis of the corona virus (covid-19 pandemic). This paper describes and analyzes Intelligent Tutoring Systems as a solution to the problem of availability, efficiency, lifelong-higher education in a theoretical way with the help of conducted studies by some of the researchers with the aim of promoting Intelligent Tutoring Systems and non-use of Intelligent Tutoring Systems in higher education institutions. Goal is to explore and describe the stages of their development and the main models from which the architecture is made, how programming classes are started once they are implemented on the basis of studies by some of the researchers, and identify their advantages and disadvantages. Structures and how they work or learning in some of the intelligent tutoring systems for programming were mentioned and described. It's also presented intelligent tutoring system for teaching programming using dynamic graphic visualizations with an example. Future developments have been proposed with proposals to improve the current ones. But the implementation of Intelligent Tutoring Systems in educational institutions is for now a future, as 1 hour of classroom instruction requires hundreds of hours of programming for 1 hour of teaching through Intelligent Tutoring System. But when implemented, we could fundamentally change the way we know it. As for learning programming, an intelligent tutoring system for learning programming could even come to life sooner than in other areas. The potential impact has yet to be seen, but it is considered clear that the future of Intelligent Tutoring Systems is bright.

Key words: Intelligent tutoring systems, programming, teaching

LITERATURA

21KSchool, „Exploring the Role of Intelligent Tutoring Systems in Education“, internet, link: https://www.21kschool.world/ge/blog/intelligent-tutoring-systems/?fbclid=IwAR1aR8dAsj_bOzjiOOOZTwyix8l5-5v-00exlGVHZxkbKnzi6LGQfc2wz5g#Pros_and_Cons_of_Intelligent_Tutoring_Systems pristupljeno (20.09.2023)

Abdolhossein Sarrafzadeh, Sam Alexander, Farhad Dadgostar, Chao Fan, Abbas Bigdeli, „“How do you know that I don't understand?” A look at the future of intelligent tutoring systems“, internet, link: https://www.researchgate.net/publication/223932101_How_do_you_know_that_I_do_n't_understand_A_look_at_the_future_of_intelligent_tutoring_systems pristupljeno (20.09.2023)

Akyuz Yasar, „Effects of Intelligent Tutoring Systems (ITS) on Personalized Learning (PL)“, internet, link: https://www.researchgate.net/publication/342525969_Effects_of_Intelligent_Tutoring_Systems_ITS_on_Personalized_Learning_PL pristupljeno (20.09.2023)

Albert T. Corbett, John R. Anderson, "Student Modelling and Mastery Learning in a Computer-Based Programming Tutor", internet, link: https://link.springer.com/chapter/10.1007/3-540-55606-0_49 pristupljeno (20.09.2023)

Amy Ogan, Samantha Finkelstein, Erin Walker, Ryan Carlson, Justine Cassell, "Rudeness and Rapport: Insults and Learning Gains in Peer Tutoring", internet, link: https://link.springer.com/chapter/10.1007/978-3-642-30950-2_2 pristupljeno (20.09.2023)

Ani Amižić, Slavomir Stankov, Marko Rosić, „Model traganja – dijagnostička tehnika inteligentnih tutorskih sustava“, internet, link: https://www.academia.edu/29531605/Model_traganja_dijagnosti%C4%8Dka_tehnika_inteligentnih_tutorskih_sustava pristupljeno (20.09.2023)

Antonija Mitrovic, Stellan Ohlsson, „Constraint-based knowledge representation for individualized instruction“, internet, link:

<https://ir.canterbury.ac.nz/server/api/core/bitstreams/9d8a6e5b-de67-4b3f-97d9-64bbf5375242/content>

pristupljeno (20.09.2023)

Antonija Mitrovic, Stellan Ohlsson, „Evaluation of a constraint-based tutor for a database language“, internet, link:

https://www.researchgate.net/publication/29486968_Evaluation_of_a_constraint-based_tutor_for_a_database_language

pristupljeno (20.09.2023)

Antonija Mitrovic, Stellan Ohlsson, Devon K. Barrow, „The effect of positive feedback in a constraint-based intelligent tutoring system“, internet, link:

<https://www.sciencedirect.com/science/article/abs/pii/S0360131512001613>

pristupljeno (20.09.2023)

Antonija Mitrović, „An intelligent SQL tutor on the web“, internet, link:

https://scholar.google.com/citations?view_op=view_citation&hl=en&user=qlf7b10AAA&AJ&citation_for_view=qlf7b10AAAAJ:d1gkVwhDpl0C

pristupljeno (20.09.2023)

Arthur C. Graesser, Kurt VanLehn, Carolyn Penstein Rosé, Pamela W. Jordan, Derek Harter, „Intelligent tutoring systems with conversational dialogue“, internet, link:

https://www.researchgate.net/publication/220017573_Intelligent_Tutoring_Systems_with_Conversational_Dialogue

pristupljeno (20.09.2023)

Arthur C. Graesser, Patrick Chipman, Brian C. Haynes, Andrew Olney, „AutoTutor: An intelligent tutoring system with mixed-initiative dialogue“, internet, link:

https://www.researchgate.net/publication/3051047_AutoTutor_An_Intelligent_Tutoring_System_With_Mixed-Initiative_Dialogue

pristupljeno (20.09.2023)

Bashar G H Al-Bastami, Samy S. Abu-Naser, „Design and Development of an Intelligent Tutoring System for C# Language“, internet, link: https://www.researchgate.net/publication/314229313_Design_and_Development_of_an_Intelligent_Tutoring_System_for_C_Language Design and Development of an Intelligent Tutoring System for C Language
pristupljeno (20.09.2023)

Binh Hoang Tieu, The Duy Bui, „Responsive student model in an intelligent tutoring system and its evaluation“, internet, link: https://www.researchgate.net/publication/350495512_Responsive_student_model_in_an_intelligent_tutoring_system_and_its_evaluation
pristupljeno (20.09.2023)

Chris Sadler, „The Future of AI Tutoring in Higher Ed“, internet, link: https://www.newamerica.org/oti/briefs/the-future-of-ai-tutoring-in-higher-ed/?fbclid=IwAR16k8rrdft2j5e_ICARsN4eA1rINK1x-GBq9kswv-yHuTyUlfWvrHcnXHs
pristupljeno (20.09.2023)

Cory J. Butz, Shan Hua, Brien Maguire, „A Web-Based Intelligent Tutoring System for Computer Programming“, internet, link: https://www.researchgate.net/publication/221158161_A_Web-Based_Intelligent_Tutoring_System_for_Computer_Programming?fbclid=IwAR20E9_ZL5VDu3U1xqKGx1rruDOXdEmzilOHNoJhCpQaW0EFLvyyVTnWaTaY
pristupljeno (20.09.2023)

Daniel Weitekamp, Erik Harpstead i Ken R. Koedinger, „An Interaction Design for Machine Teaching to Develop AI Tutors“, internet, link: https://dl.acm.org/doi/abs/10.1145/3313831.3376226?fbclid=IwAR3_7YRxK6AWETR_6rdXMMmlqTWxzmVKeXTGIH6Jf7L0hCEIa8OdsCbK3F_4
pristupljeno (20.09.2023)

Enrique Sierra, Alejandro Hossian, Paola Britos, Dario Rodriguez, Ramon Garcia-Martinez, "A Multi-agent Intelligent Tutoring System for Learning Computer Programming", internet, link: <https://ieeexplore.ieee.org/abstract/document/4367717>
pristupljeno (20.09.2023)

Carolina Gonzalez, Juan Carlos Burguillo, Martín Llamas Nistal, Rosalía Laza Fidalgo, „Designing Intelligent Tutoring Systems: A Personalization Strategy using Case-Based Reasoning and Multi-Agent Systems“, internet, link: https://www.researchgate.net/publication/307642038_Designing_Intelligent_Tutoring_Systems_A_Personalization_Strategy_using_Case-Based_Reasoning_and_Multi-Agent_Systems

pristupljeno (20.09.2023)

Hyacinth S. Nwana, „Intelligent tutoring systems: an overview“, internet, link: <https://link.springer.com/article/10.1007/BF00168958#citeas>

pristupljeno (20.09.2023)

John R. Anderson, C. Franklin Boyle, Brian J. Reiser, „Intelligent tutoring systems“, internet, link: <https://www.science.org/doi/10.1126/science.228.4698.456>

pristupljeno (20.09.2023)

John R. Anderson, Albert T. Corbett, Kenneth R. Koedinger, „Intelligent tutoring systems, internet, link: http://act-r.psy.cmu.edu/wordpress/wp-content/uploads/2012/12/173Chapter_37_Intelligent_Tutoring_Systems.pdf

pristupljeno (20.09.2023)

Kenneth R. Koedinger, John R Anderson, William H. Hadley, Mary A. Mark, „Intelligent Tutoring Goes To School in the Big City“, internet, link: https://www.researchgate.net/publication/2834538_Intelligent_Tutoring_Goes_To_School_in_the_Big_City

pristupljeno (20.09.2023)

Matthew Lynch, „The benefits and limitations of machine learning in education“, internet, link: <https://www.thetechadvocate.org/the-benefits-and-limitations-of-machine-learning-in-education/>

pristupljeno (20.09.2023)

Meng Wang, Wenjun Wu, Yu Liang, „A Novel Intelligent Tutoring System For Learning Programming“, internet, link: https://www.researchgate.net/publication/341948530_A_Novel_Intelligent_Tutoring_System_For_Learning_Programming

pristupljeno (20.09.2023)

Mohammed Al-Shawwa, Izzeddin Alshawwa, „ITS for Learning Java“, internet, link: <http://dstore.alazhar.edu.ps/xmlui/bitstream/handle/123456789/574/ITS-IJAISR190101-k.pdf?sequence=1&isAllowed=y>

pristupljeno (20.09.2023)

Ning Wang, Jonathan Gratch, „Rapport and facial expression“, internet, link: https://www.researchgate.net/publication/224088146_Rapport_and_facial_expressio_n

pristupljeno (20.09.2023)

Paula Valverde, „Revolutionising learning: Intelligent Tutoring Systems powered by Artificial Intelligence“, internet, link: https://theintelligencerevolution.com/revolutionising-learning-intelligent-tutoring-systems-powered-by-artificial-intelligence/?fbclid=IwAR3ublZSVZtmp4X3sy7JYOI7WN4hqr5laIJ8r_sheSDogKm6mf7ikviolgl

pristupljeno (20.09.2023)

Peter Sedlmeier, „Intelligent Tutoring Systems“, internet, link: https://www.researchgate.net/publication/291600956_Intelligent_Tutoring_Systems

pristupljeno (20.09.2023)

Ping Mu, „Research on Artificial Intelligence Education and Its Value Orientation“, internet, link: https://webofproceedings.org/proceedings_series/ESSP/IETRC%202019/IETRC19165.pdf

pristupljeno (20.09.2023)

Prynka Sharma, Mayuri Harkishan, „Designing an intelligent tutoring system for computer programing in the Pacific“, internet, link: <https://doi.org/10.1007/s10639-021-10882-9>

pristupljeno(20.09.2023)

Roger Nkambou, Riichiro Mizoguchi, Jacqueline Bourdeau, „Advances in Intelligent Tutoring Systems“, internet, link: https://www.researchgate.net/publication/268992454_Advances_in_Intelligent_Tutori_ng_Systems

pristupljeno (20.09.2023)

Samy S. Abu Naser, „Developing an Intelligent Tutoring System For Students Learning To Program in C++“, internet, link: <https://scialert.net/abstract/?doi=itj.2008.1055.1060> pristupljeno (20.09.2023)

Santiago Schez-Sobrinio, Cristian Gmez-Portes, David Vallejo, Carlos Glez-Morcillo, Miguel Á. Redondo, „An Intelligent Tutoring System to Facilitate the Learning of Programming through the Usage of Dynamic Graphic Visualizations“, internet, link: <https://www.mdpi.com/2076-3417/10/4/1518> pristupljeno (20.09.2023)

S. H. Hahn , J. S. Song, K. Y. Tak, J. H. Kim, „An Intelligent Tutoring System for Introductory C language Course“, internet, link: <https://citeseerx.ist.psu.edu/document?repid=rep1&type=pdf&doi=947f453be7254b18d3920cb77264278c49974298> pristupljeno (20.09.2023)

Sidney D'Mello, Art Graesser, „Dynamics of affective states during complex learning“, internet, link: https://www.researchgate.net/publication/251624186_Dynamics_of_affective_states_during_complex_learning pristupljeno (20.09.2023)

Sidney D'Mello, Andrew Olney, Claire Williams, Patrick Hays, „Gaze tutor: A gaze-reactive intelligent tutoring system“, internet, link: https://www.researchgate.net/publication/257481863_Gaze_tutor_A_gaze-reactive_intelligent_tutoring_system pristupljeno (20.09.2023)

Slavomir Stankov, „Inteligentni tutorski sustavi“, internet, link: http://slavomirstankov.com/wordpress/wp-content/uploads/2014/07/Dio_C.pdf pristupljeno (20.09.2023)

Stellan Ohlsson, „Constraint based student modeling“, internet, link: https://scholar.google.com/citations?view_op=view_citation&hl=en&user=qdZZe9IAAAAJ&citation_for_view=qdZZe9IAAAAJ:ufrVoPGSRksC pristupljeno (20.09.2023)

Stellan Ohlsson, „Learning from performance errors“, internet, link: <https://psycnet.apa.org/doiLanding?doi=10.1037%2F0033-295X.103.2.241>
pristupljeno (20.09.2023)

Tom Murray, „AN OVERVIEW OF INTELLIGENT TUTORING SYSTEM AUTHORIZING TOOLS: Updated Analysis of the State of the Art“, internet, link: https://d1wqtxts1xzle7.cloudfront.net/88633341/978-94-017-0819-720220715-1-15x9vrm-libre.pdf?1657918919=&response-content-disposition=inline%3B+filename%3DAuthoring_Tools_for_Advanced_Technology.pdf&Expires=1695682207&Signature=RjeSqRO~9GJvGUZ4rzeKmsGnAgjATatIssraN8-nrsQZEKBFyGBud87Y8Nfd8EMWYvdPLtgYNEHp5ggwfbG~EISYAyUq2BWYycufQMdzVXVUBw-KiJWfJ-FglBY-DXf-efxsXeb6j5ojpqHn-Kh07EDYysCX7XYtyj1yfrzc-ACt2s9aV2kKb2-jF9YEEERx3wUUbqHZk~WRyYM3xDA~9VxpqwNHg0UWbGjKdmJt~w~Aqm7iH5nDKcQ-ijl4lI3Jq29sSN9mICiuRcZapdpPdhinejF4oHb~zR3CSbl0F07WnTiCkEjVESLsOb80mrZ21M0XooMTBjYbTrppLfkc2cA_&Key-Pair-Id=APKAJLOHF5GGSLRBV4ZA#page=497
pristupljeno (20.09.2023)

Tonči Dadić, „Inteligentni tutorski sustavi za poučavanje programiranja“, Internet, link: https://bib.irb.hr/datoteka/381144.Dadic_DKI_19062008.pdf
pristupljeno (20.09.2023)

University of Denver Bootcamps, „18 Skills All Programmers Need to Have“, internet, link: <https://bootcamp.du.edu/blog/programming-skills/>
pristupljeno (20.09.2023)

Yorah Bosse, Marco Aurelio Gerosa, „Why is programming so difficult to learn?: Patterns of Difficulties Related to Programming Learning Mid-Stage“, internet, link: https://www.researchgate.net/publication/312142448_Why_is_programming_so_difficult_to_learn_Patterns_of_Difficulties_Related_to_Programming_Learning_Mid-Stage
pristupljeno (20.09.2023)

POPIS SLIKA

Slika 1 Prikaz tradicionalne arhitekture modela traganja	9
Slika 2. Arhitektura „Novel Intelligent tutoring system for learning programing“	15
Slika 3. Komponente „Novel Intelligent tutoring system-a“	16
Slika 4. Primjer davanja savjeta i popravka greške u „Novel Intelligent tutoring system“	17
Slika 5 Primjer zadatka s rješenjem i nekoliko mogućih netočnih rješenja iz CPP tutora	19
Slika 6. Dijagram toka za CPP tutor	21
Slika 7. Primjer uključivanja CPP tutora u dijalog s učenicom	22
Slika 8 Arhitektura ITS-a za učenje C# programskog jezika	24
Slika 9. Login screen Inteligentnog tutorskog sustava za učenje C# programskog jezika	27
Slika 10. Biranje lekcije koju učenik želi učiti iz Inteligentnog tutorskog sustava za učenje C# programskog jezika	27
Slika 11. Biranje primjera iz lekcije koju učenik želi učiti iz Inteligentni tutorski sustav za učenje C# programskog jezika.....	28
Slika 12. Pitanja za korisnika nakon čitanja lekcije iz Inteligentni tutorski sustav za učenje C# programskog jezika	29
Slika 13. Evaluacija odgovora nakon pitanja za korisnika nakon čitanja lekcije iz Inteligentni tutorski sustav za učenje C# programskog jezika.....	29
Slika 14. Primjer dodavanja novog studenta i upis podataka o njemu iz Inteligentni tutorski sustav za učenje C# programskog jezika.....	30
Slika 15. Primjer izgleda sučelja o statusu, razini pitanja, problema i ostvarenih bodova iz Inteligentni tutorski sustav za učenje C# programskog jezika	31
Slika 16. Primjer izgleda pop up prozora s savjetom tutora iz Inteligentni tutorski sustav za učenje C# programskog jezika.....	31
Slika 17. Opći pregled predloženog inteligentnog tutorskog sustava.....	32

Slika 18. Grafički prikazi notacije prometnih znakova za olakšavanje učenja programiranja pretvoreni u ortogonalno projicirani 3D prostor.....	33
Slika 19. Generiranje JSON strukture s relevantnim informacijama koje se koriste za izgradnju vizualizacije iz izvornog koda programa.....	34
Slika 20. Dinamička vizualizacija algoritma “Bubble sort” viđena kroz AR uređaj....	35